# Red Hat JBoss Operations Network 3.3
# Installation Guide

---

For All Server and Agent Installation Procedures and Guidelines

Jared Morgan Zach Rhoads Ella Deon Ballard

# Red Hat JBoss Operations Network 3.3 Installation Guide

## For All Server and Agent Installation Procedures and Guidelines

Jared Morgan
jmorgan@redhat.com

Zach Rhoads
zach@redhat.com

Ella Deon Ballard
dlackey@redhat.com

## Legal Notice

Copyright © 2015 Red Hat.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at
http://creativecommons.org/licenses/by-sa/3.0/
. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, MetaMatrix, Fedora, the Infinity Logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack Logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

## Abstract

This manual covers the installation and setup of JBoss ON 3.3 servers and agents and basic tasks for configuring the inventory.

# Table of Contents

**Preface** ..... **4**
1. JBoss Operations Network Overview 4
2. Examples and Formatting 4
3. Giving Feedback 6

**Chapter 1. Setting up Databases** ..... **7**
1. Configuring PostgreSQL 7
2. Setting up Oracle 10

**Chapter 2. Installing the JBoss ON Server** ..... **16**
1. Supported Platforms, Databases, and Other Requirements 16
2. Hardware Minimums 16
3. Disk Space Considerations 16
4. Preparing for Installation on *nix Systems 17
5. Preparing for Installation on Windows 19
6. About the rhqctl Script 25
7. Basic Setup: Installing the Server on Linux 30
8. Basic Setup: Installing the Server on Windows 32
9. Installing Additional Servers for High Availability 35
10. Installing Storage Nodes Before Installing the Server 36
11. Managing the Server Service 38

**Chapter 3. Upgrading JBoss ON Servers and Storage Nodes** ..... **40**
1. Upgrade Notes 40
2. About the Upgrade Script 41
3. Upgrading a 3.1.x Server and Server Plug-ins 42
4. Upgrading a 3.2 Server, Storage Nodes, and Server Plug-ins 47
5. Re-Installing the Server 51

**Chapter 4. Uninstalling the JBoss ON Server** ..... **52**
1. Uninstalling an Agent on a Managed System 52
2. Uninstalling the Server 54

**Chapter 5. Installing and Upgrading an Agent on a Managed Platform from the JAR File** ..... **55**
1. Before Installing the Agent 55
2. Installing the Agent from JAR File 58
3. Silently Installing an Agent 62
4. Running the JBoss ON Agent as a Service 64
5. Changing Agent Connection Configuration 68
6. About Agent Automatic Updates 68
7. Manually Upgrading the JBoss ON Agent 72
8. Reinstalling the Agent 73
9. Starting the Agent 74

**Chapter 6. Installing the Agent from RPM** ..... **75**
1. About Agent RPMs 75
2. Installing the Agent from RPM 79
3. Changing the Agent Configuration After an RPM Install 84
4. Migrating from a JAR Installation to an RPM Installation 85
5. Starting the Agent 88
6. Upgrading the Agent RPM 88
7. Troubleshooting RPM Installs 92

**Chapter 7. Installing an Agent Using the JBoss ON User Interface** **93**

**Chapter 7. Installing an Agent Using the JBoss ON User Interface** ..... **93**
1. Installation Options 94

**Chapter 8. Installing and Removing JBoss Agent Plug-in Packs** ..... **96**
1. Installing JBoss Agent Plug-in Packs 96
2. Removing JBoss Agent Plug-in Packs 97

**Chapter 9. Installing the JBoss ON CLI** ..... **99**

**Chapter 10. Troubleshooting Installation and Upgrade** ..... **100**
10.1. Exceptions and Error Logs 100
10.2. Connection Issues 101

**Appendix A. Document History** ..... **103**

# Preface

JBoss Operations Network 3.3 provides an integrated solution for managing JBoss middleware, other network infrastructure, and applications built on Red Hat Enterprise Application Platform (EAP).

This manual covers planning and procedures for installing JBoss ON servers and agents and upgrading existing JBoss ON systems. This *Installation Guide* is intended for JBoss ON administrators.

Report a bug

## 1. JBoss Operations Network Overview

JBoss Operations Network has four major components, which work together to create the management platform:

- The JBoss ON servers, which centralize configuration and connect the components
- An SQL database (PostgreSQL or Oracle) which stores JBoss ON configuration settings and resource-related data, including content packages, the resource inventory, and monitoring data
- Local agents installed on managed platforms, which connect with servers to receive resource configuration updates and which collect and send monitoring data
- The JBoss ON GUI, which is a web-based interface that allows users to connect to any JBoss ON server, from any location, to view resource data and perform management tasks

Report a bug

## 2. Examples and Formatting

Each of the examples used in this guide, such as file locations and commands, have certain defined conventions.

Report a bug

### 2.1. Command and File Examples

All of the examples for JBoss ON commands, file locations, and other usage are given for Red Hat Enterprise Linux systems. Be certain to use the appropriate commands and files for your platform.

**Example 1. Example Command**

To start the JBoss ON server:

```
serverRoot/jon-server-3.3.2.GA/bin/rhqctl start
```

Report a bug

## 2.2. Text Formatting and Styles

Certain words are represented in different fonts, styles, and weights. Different character formatting is used to indicate the function or purpose of the phrase being highlighted.

| Formatting Style | Purpose |
|---|---|
| **Monospace font** | Monospace is used for commands, packa files and directory paths, and any text dis prompt. |
| `Monospace with a background` | This type of formatting is used for anythin or returned in a command prompt. |
| *Italicized text* | Any text which is italicized is a variable, s *instance_name* or *hostname*. Occasionally, used to emphasize a new term or other ph |
| **Bolded text** | Most phrases which are in bold are appli names, such as **Cygwin**, or are fields or user interface, such as a **User Name He** **Save** button. |

Other formatting styles draw attention to important text.

**NOTE or TIP**

A note provides additional information that can help illustrate the behavior of the system or provide more detail for a specific issue. Tips provide pointers to helpful information or to easy ways to accomplish something.

**Important**

Important information is necessary, but possibly unexpected, such as a configuration change that will not persist after a reboot.

**Warning**

A warning indicates potential data loss, as may happen when tuning hardware for maximum performance.

Report a bug

## 3. Giving Feedback

If there is an error in this guide, or there are ways to improve the documentation, please let us know. Bugs can be filed against the documentation for the product in Bugzilla, http://bugzilla.redhat.com/bugzilla. Make the bug report as specific as possible, so we can be more effective in correcting any issues:

1. Select the **JBoss** products group.
2. Select **Red Hat JBoss Operations Network** from the list.
3. Set the component to **Documentation**.
4. Set the version number to 3.3.
5. For errors, give the page number (for the PDF) or URL (for the HTML), and give a succinct description of the problem, such as incorrect procedure or typo.

   For enhancements, put in what information needs to be added and why.
6. Give a clear title for the bug. For example, **"Incorrect command example for setup script options"** is better than **"Bad example"**.

We appreciate receiving any feedback — requests for new sections, corrections, improvements, enhancements, even new ways of delivering the documentation or new styles of docs. You are welcome to contact Red Hat Customer Content Services directly through the Authors of this document.

Report a bug

# Chapter 1. Setting up Databases

## 1. Configuring PostgreSQL

Running JBoss Operations Network on PostgreSQL requires three things:

- Adequate PostgreSQL settings for memory, timeouts, connections, and related settings
- A database
- A user with adequate permissions

JBoss ON supports PostgreSQL 8.4.x, 9.0.x, and 9.1.x.

Report a bug

### 1.1. Configuring PostgreSQL

> **Important**
>
> The following configuration is provided as an example of configuring this server quickly for a JBoss ON testing environment. Suggested values in these procedures should not be used in production environments. The procedure should not be used as a supported way of configuring a production server. Always follow the database provider configuration instructions carefully when configuring a production environment.

For detailed information about setting up client authentication for PostgreSQL users and databases, see the PostgreSQL documentation for the supported version at http://www.postgresql.org/docs/8.4/interactive/client-authentication.html.

> **Note**
>
> Ensure that the Postgres authentication mechanism is properly configured for the configuration commands to work.

1. *Optional.* Change the password for the Unix user for PostgreSQL:

   ```
   sudo passwd postgres
   ```

2. Initialize the database. The database must be initialized before starting the server.

   ```
   service postgresql initdb
   ```

3. Start Postgres. For example, on Red Hat Enterprise Linux:

   -

```
service postgresql start
```

On Windows:

```
net start pgsql-8.4
```

4. Set up a password for the **postgres** user on the database:

```
# su - postgres
$ psql
postgres=# ALTER USER postgres PASSWORD 'password';
ALTER ROLE
postgres=#
```

5. Create a PostgreSQL role named **rhqadmin** with password **rhqadmin**.

```
postgres=# CREATE USER rhqadmin PASSWORD 'rhqadmin';
CREATE ROLE
```

6. Create a PostgreSQL database named **rhq**, specifying the **rhqadmin** role as the owner.

```
postgres=# CREATE DATABASE rhq OWNER rhqadmin;
CREATE DATABASE
```

7. Give users on the computer access to the database. To allow all users, add the appropriate connection settings for each connection type (local, IPv4, and IPv6) to the **data/pg_hba.conf** configuration file, for both local and external connections:

```
# "local" is for Unix domain socket connections only
local   all         all                               md5
# IPv4 local connections:
host    all         all         127.0.0.1/32           md5
host    all         all         172.31.7.0/24          md5
# IPv6 local connections:
host    all         all         ::1/128               md5
```

Using **all all** sets these settings for every user to every PostgreSQL database. This settings can be applied to only the JBoss ON database by using **rhq all** or even to specific users for JBoss ON, such as **rhq rhqadmin**.

Then, restart the database service.

```
service postgresql restart
```

8. Make the configuration changes in Section 1.2, “Setting PostgreSQL Parameters”.

Report a bug

## 1.2. Setting PostgreSQL Parameters

There are several settings in the PostgreSQL server configuration that can be tuned to provide better performance for JBoss ON.

Report a bug

### 1.2.1. Editing the postgresql.conf File

PostgreSQL requires minor changes to the database configuration in the **postgresql.conf** file.

1. Make sure that an adequate amount of memory and system resources are assigned to accommodate the JBoss ON database.

```
## not necessary  if the database is started with the -i flag
listen_addresses = '*'

## performance changes for JBoss ON
shared_buffers = 80MB   #  default is 32MB
work_mem = 2048   #  default is 1MB
checkpoint_segments = 10 #  default is 3
```

**Note**

The parameter ***statement_timeout*** should not be set. If **postgressql.conf** contains a statement_timeout parameter, it should be overridden for the JBoss ON database user:

```
ALTER USER rhqadmin SET statement_timeout=0;
```

Report a bug

### 1.2.2. Setting Kernel Parameters

Consider adjusting the kernel parameters for your system. The PostgreSQL documentation on Managing Kernel Resources has more information.

Report a bug

### 1.2.3. Editing pg_hba.conf

Update the **pg_hba.conf** file to allow the newly-created role to connect from the machine the JBoss ON server is installed on, such as localhost. Adding client connections is covered in the PostgreSQL documentation in the Client Authentication section.

After editing the **pg_hba.conf** file, restart PostgreSQL for the changes to take effect. If no errors are displayed, the database is now ready to support a JBoss ON installation.

For more information on tuning Postgres, see the PostgreSQL documentation about Tuning your PostgreSQL Server.

Report a bug

### 1.2.4. Fixes for "Relation RHQ_Principal does not exist" Error

Sometimes the database connection is marked as valid but the install still fails with the *Relation RHQ_Principal does not exist* error. This occurs when a new database is created by running **initdb** in a ***non-C*** locale through PostgreSQL instances.

To fix this error:

1. Using a database explorer, create an empty table called ***RHQ_PRINCIPAL*** in the database used for JBoss ON.

2. Click **Install server**.

   The installer displays a warning about an existing schema. Overwrite the existing schema as it only consists of one empty table.

Another option is to specify the encoding of the created database as ***SQL-ASCII*** at creation time. For example:

```
initdb -D /my/test/data -E SQL_ASCII --locale en_US.UTF-8
```

Report a bug

## 2. Setting up Oracle

Only two things are required to run JBoss ON on Oracle:

- A database
- A user with adequate permissions

Basic configuration follows the process of setting up the database and users. There is also an advanced configuration process that gives more control over the database settings, such as increased memory limits, that can improve performance for large JBoss ON deployments.

Report a bug

### 2.1. Prepping Oracle Settings

There are several settings in the Oracle configuration that can be tuned to provide better performance for JBoss ON.

Report a bug

### 2.1.1. Setting SGA and PGA Sizes

Oracle settings for SGA and PGA sizes are very important for JBoss ON performance. If these values are too small, the database will be very slow. There are two specific settings to adjust:

- sga_target
- pga_aggregate_target

Talk to the database administrator to verify the sizing requirements for Oracle's SGA and PGA settings.

Report a bug

### 2.1.2. Tuning Open Cursors

Run the following SQL command to check if the ***max_open_cur*** setting has a value lower than 300:

```
select max(a.value) as highest_open_cur, p.value as max_open_cur
from v$sesstat a, v$statname b, v$parameter p
where a.statistic# = b.statistic#
and b.name = 'opened cursors current'
and p.name= 'open_cursors'
group by p.value;
```

If the value *is* lower then 300, then open more cursors:

```
alter system set open_cursors=300 scope=spfile;
```

**Note**

This query applies only to the existing sesssion. When the session is disconnected, the setting is returned to it's previous value.

Report a bug

### 2.1.3. Setting the Number of Processes and Sessions

The **v$resource_limit** limit sets the maximum number of Oracle processes and sessions

which JBoss ON is allowed to have. The equation for this calculation has this general flow:

```
calculate the number of processes => add additional processes for
Enterprise Manager => calculate the total number of sessions (final
value)
```

There are two ways to calculate the number of processes (one using the number of agents and the other the number of servers). Use whichever method results in a higher number.

**Table 1.1. Calculating Oracle Processes**

| Calculation Type | Equation | Example |
|---|---|---|
| Agents | 1.5 * *number_of_agents* | 1.5 * 100 agents = 150 |
| Servers | 60 * *number_of_servers* | 60 * 2 servers = 120 |
| with Oracle Enterprise Manager | *highest_number_of_processes* + 40 | 1.5 * 100 agents + 40 = 190 |

As noted in Table 1.1, “Calculating Oracle Processes”, the calculation is slightly different for systems using Oracle Enterprise Manager. In that situation, first calculate the processes for agents and servers. Then, take whichever value is highest and add another 40, and that yields the number of processes to set.

After calculating the total number of processes, then take that number and multiply it by 1.1 to determine the total number of sessions (and the final value for **`v$resource_limit`**).

**Example 1.1. Calculating Oracle Processes and Sessions for JBoss ON**

Example Corp. is planning to deploy 175 agents and 3 servers. They will be using Oracle Enterprise Manager to manage their Oracle instance.

The first step is to calculate the number of processes based on agents and based on servers:

```
1.5 * 175 agents = 262.5 processes
60 * 3 servers = 180 process
```

So the method to use for processes is the agent's method, since that value is higher.

They add another 40 to the number of processes to accommodate the Oracle Enterprise Manager.

```
262.5 + 40 = 302.5
```

The total number of process is 302.5. From there, they calculate the number of sessions:

```
302.5 * 1.1 = 332.75
```

The final value for their Oracle **v$resource_limit** limit database setting is 333.

Report a bug

## 2.2. Configuring Oracle

**Important**

The following configuration is provided as an example of configuring this server quickly for a JBoss ON testing environment. Suggested values in these procedures should not be used in production environments. The procedure should not be used as a supported way of configuring a production server. Always follow the database provider configuration instructions carefully when configuring a production environment.

A specific Oracle database and user need to be configured for JBoss ON to access to store its data.

1. Create a dedicated Oracle instance to be used for JBoss ON. This process is described in the Oracle documentation.

2. Log into Oracle as the system user.

```
[jsmith@server ~]$ sqlplus
SQL> CONNECT sys/your_sys_password AS sysdba;
```

3. Create a database for JBoss ON. In this example, the database is named **rhq**. This process is described in more detail in the Oracle documentation.

```
SQL> CREATE DATABASE rhq;
SQL> @?/rdbms/admin/catalog.sql
SQL> @?/rdbms/admin/catproc.sql
```

4. Create a user that JBoss ON will use to access Oracle. Create the user named **rhqadmin** with the password **rhqadmin**. For example:

```
SQL> CREATE USER rhqadmin IDENTIFIED BY rhqadmin;
```

5. Grant the required permissions to the Oracle user. At a minimum, this user must have the **connect** and **resource** roles. For example:

```
SQL> GRANT connect, resource TO rhqadmin;
```

**Important**

When configuring Oracle 12c as a backend for JBoss ON, note that the ***RESOURCE*** role no longer grants ***UNLIMITED TABLESPACE*** to the **rhqadmin** user by default. Granting this system privilege to a user must be done manually.

To accomplish this, the **rhqadmin** user must have adequate ***QUOTA*** for the necessary tablespaces.

If granting unlimited tablespace is acceptable, then this can be done explicitly:

```
GRANT UNLIMITED TABLESPACE TO rhqadmin;
```

Otherwise, specific quota limits must be set. For example:

```
ALTER USER $username QUOTA 100G ON $tablespace_name;
```

Or set it to unlimited:

```
ALTER USER $username QUOTA -1 ON $tablespace_name;
```

6. Set additional permissions for the JBoss ON Oracle user that define parameters to handle database commits.

   JBoss ON uses internally two phase commit for some of database actions. To recover from two phase commit failures, the Oracle user has to has appropriate permissions, otherwise the database will return **XAException.XAER_RMERR** errors.

   Set these four privileges for the user:

   ```
   GRANT SELECT ON sys.dba_pending_transactions TO user;
   GRANT SELECT ON sys.pending_trans$ TO user;
   GRANT SELECT ON sys.dba_2pc_pending TO user;
   GRANT EXECUTE ON sys.dbms_xa TO user;
   ```

   The **GRANT EXECUTE** line assumes that the Oracle server is version 11g R1. For an unpatched version of Oracle older than 11g R1, then use this line instead:

   ```
   GRANT EXECUTE ON sys.dbms_system TO user;
   ```

7. Make sure that the **db_block_size** value is at least 8 KB.

   ```
   SQL> show parameter db_block_size;
   NAME                                 TYPE        VALUE
   ------------------------------------ ----------- ------------
   ------------------
   db_block_size                        integer     8192
   ```

Report a bug

# Chapter 2. Installing the JBoss ON Server

The core of JBoss Operations Network is the server, which communicates with agents, maintains the inventory, manages resource settings, interacts with content providers, and provides a central management UI. JBoss ON has other components which are required in order for JBoss ON to carry out its functions — agents which are installed on platforms, a CLI which allows administrators to script configuration, and plug-ins which integrate JBoss ON with other JBoss products. Each component has to be installed and configured independently, to match the needs of the specific network.

Report a bug

## 1. Supported Platforms, Databases, and Other Requirements

The list of supported platforms, databases, and other requirements such as Java, are listed at https://access.redhat.com/knowledge/articles/112523.

Report a bug

## 2. Hardware Minimums

Regardless of the server or database platform, there are certain minimum requirements that must be met to install the JBoss ON server and its associated database.

**Table 2.1. Recommended Minimum Hardware**

| | Minimum |
|---|---|
| Memory | 2 GB |
| Installation Directory Storage [a] | 10 GB |
| Temporary Directory Storage | 10 GB |

[a] The server runs as a system user. Make sure that any system limits on user memory are set high enough to accommodate the JBoss ON server and all its data.

Report a bug

## 3. Disk Space Considerations

Certain JBoss ON features can have a significant impact on storage requirements. Anything that relates to storing content in the JBoss ON database — configuration drift snapshots, bundle versions, and content-backed resources like WARs — increases the storage requirements.

**JBoss ON stores all versions of content.** Therefore, the system which hosts the backend database (Oracle or PostgreSQL) must have enough disk space to store all versions of all content for any resources using drift monitoring, content updates, and bundles. Additionally, the database itself must have adequate tablespace for the content.

When calculating the required amount of space, estimate the size of every artifact (bundle, web application, monitored directory), and then the number of versions for each artifact. At a minimum, **have twice that amount of space available**; both PostgreSQL and Oracle require twice the database size to perform cleanup operations like vacuum, compression, and backup and recovery.

Report a bug

# 4. Preparing for Installation on *nix Systems

## 4.1. Setting up the JDK for the JBoss ON Server

The JBoss ON server requires Java 6 or Java 7 JDK.

1. Download and install the appropriate version of Java, if necessary.

2. Set the **JAVA_HOME** environment variable to the installation directory.

    a. Open the **.bashrc** for the system user that will run JBoss ON. For example:

    ```
    vim /home/jon/.bashrc
    ```

    b. Add a line to set the **JAVA_HOME** environment variable to the specific JDK directory. For example:

    ```
    export JAVA_HOME=/usr/lib/jvm/java-1.6.0-openjdk-
    1.6.0.0/
    ```

3. Set the system to use the correct version of the JDK using the system **alternatives** command. The selected version has the ***+** symbols by it.

    ```
    /usr/sbin/alternatives --config javac

    There are 2 programs which provide 'javac'.

      Selection    Command
    -----------------------------------------------
       1           /usr/lib/jvm/java-1.6.0-bea/bin/javac
    ```

```
*+ 2           /usr/lib/jvm/java-1.6.0-openjdk/bin/javac

Enter to keep the current selection[+], or type selection
number:
```

Report a bug

## 4.2. Configuring NTP

**Synchronize machine clocks.** All JBoss ON servers and agents must have synchronized clocks. Clock variations cause issues in availability reporting, metric measurements, graphing, and even identifying and importing resources into inventory. The Network Time Protocol project, http://www.ntp.org/, has information on installing and configuring NTP to ensure your clocks are synchronized.

Report a bug

## 4.3. Configuring DNS

Both forward and reverse DNS mapping entries must be present for all systems for which host servers, storage nodes, and agents.

Every IP address must have a corresponding entry in the DNS server or must be explicitly defined in every **`/etc/hosts`** file for each system which is managed by JBoss ON or hosts a server or storage node.

Report a bug

## 4.4. Configuring Ports

**Configure the firewall to allow communication over the server, agent, and storage node ports.** If the required ports are blocked, then individual components will be unable to communicate with each other.

Using the default configuration, JBoss ON uses the ports listed in Table 2.2, “Default JBoss ON Ports”.

**Table 2.2. Default JBoss ON Ports**

| Port | Purpose |
|---|---|
| 7080 | Standard HTTP port for server-client communication |
| 7443 | HTTPS port for secure server-client communication |
| 16163 | For agent communication from the server |

| Port | Purpose |
|---|---|
| 9142 | For storage cluster communication |
| 7299 | For storage node JMX communication |
| 7100 | For the storage node gossip (node-to-node) communication |

Report a bug

# 5. Preparing for Installation on Windows

## 5.1. Setting up the JDK

The JBoss ON server requires Java 6 or Java 7 JDK.

If necessary, configure Windows to use the appropriate Java version.

1. Download and install the appropriate version of Java, if necessary.

2. Set the **JAVA_HOME** environment variable to the installation directory. For example:

```
C:\>set JAVA_HOME=C:\Program Files\Java\jdk1.6.0_29
```

Report a bug

## 5.2. Configuring the JVM to Run as a Service

JBoss ON includes Tanuki Software's Java service wrapper so that the JBoss ON server can be configured to run as a Windows service. Ensure that either JAVA_HOME, RHQ_JAVA_HOME, or RHQ_JAVA_EXE_FILE_PATH is set to the proper JDK or JRE.

Report a bug

## 5.3. Configuring NTP

**Synchronize machine clocks.** All JBoss ON servers and agents must have synchronized clocks. Clock variations cause issues in availability reporting, metric measurements, graphing, and even identifying and importing resources into inventory. The Network Time Protocol project, http://www.ntp.org/, has information on installing and configuring NTP to ensure your clocks are synchronized.

Report a bug

## 5.4. Configuring DNS

Both forward and reverse DNS mapping entries must be present for all systems for which host servers, storage nodes, and agents.

Every IP address must have a corresponding entry in the DNS server or must be explicitly defined in every **`/etc/hosts`** file for each system which is managed by JBoss ON or hosts a server or storage node..

Report a bug

## 5.5. Configuring Ports

**Configure the firewall to allow communication over the server, agent, and storage node ports.** If the required ports are blocked, then individual components will be unable to communicate with each other.

Using the default configuration, JBoss ON uses the ports listed in Table 2.3, "Default JBoss ON Ports".

**Table 2.3. Default JBoss ON Ports**

| Port | Purpose |
|---|---|
| 7080 | Standard HTTP port for server-client communication |
| 7443 | HTTPS port for secure server-client communication |
| 16163 | For agent communication from the server |
| 9142 | For storage cluster communication |
| 7299 | For storage node JMX communication |
| 7100 | For the storage node gossip (node-to-node) communication |

Report a bug

## 5.6. Selecting Path Names

Make sure that the *complete* path name for the server installation directory is relatively short. Path names longer than 19 characters can cause problems with executing some server tasks. Use a location such as **`C:\jon`** rather than **`C:\Documents and Settings\myusername\jon-server`**.

Also be careful when using the *extract all* command. Expanding the archive automatically creates a directory called *jon-server-VER.RELEASE/*, which is about 20 characters. Using *extract all*, instead of specifying the directory to which to extract the archive, can double the directory name by extracting to the archive name and then to a subdirectory — for example,

**C:\example\jon-server-3.3.2.GA\jon-server-3.3.2.GA**. Using other tools may install it to a downloads directory such as **C:\Users\Administrator\Downloads**.

It is recommended that you extract the archive to a short, top-level directory such as **C:\jon**, which creates an installation directory of **C:\jon\jon-server-3.3.2.GA**.

Windows' handling of file and path names is covered in the Maximum Path Length Limitation section of the *Naming Files, Paths, and Namespaces* page of the Windows Dev Center website.

Report a bug

## 5.7. Utilities to Use with JBoss ON

The only utilities used to manage the JBoss ON server are a ZIP utility to install the binaries and, possibly, a text editor to view and edit configuration files.

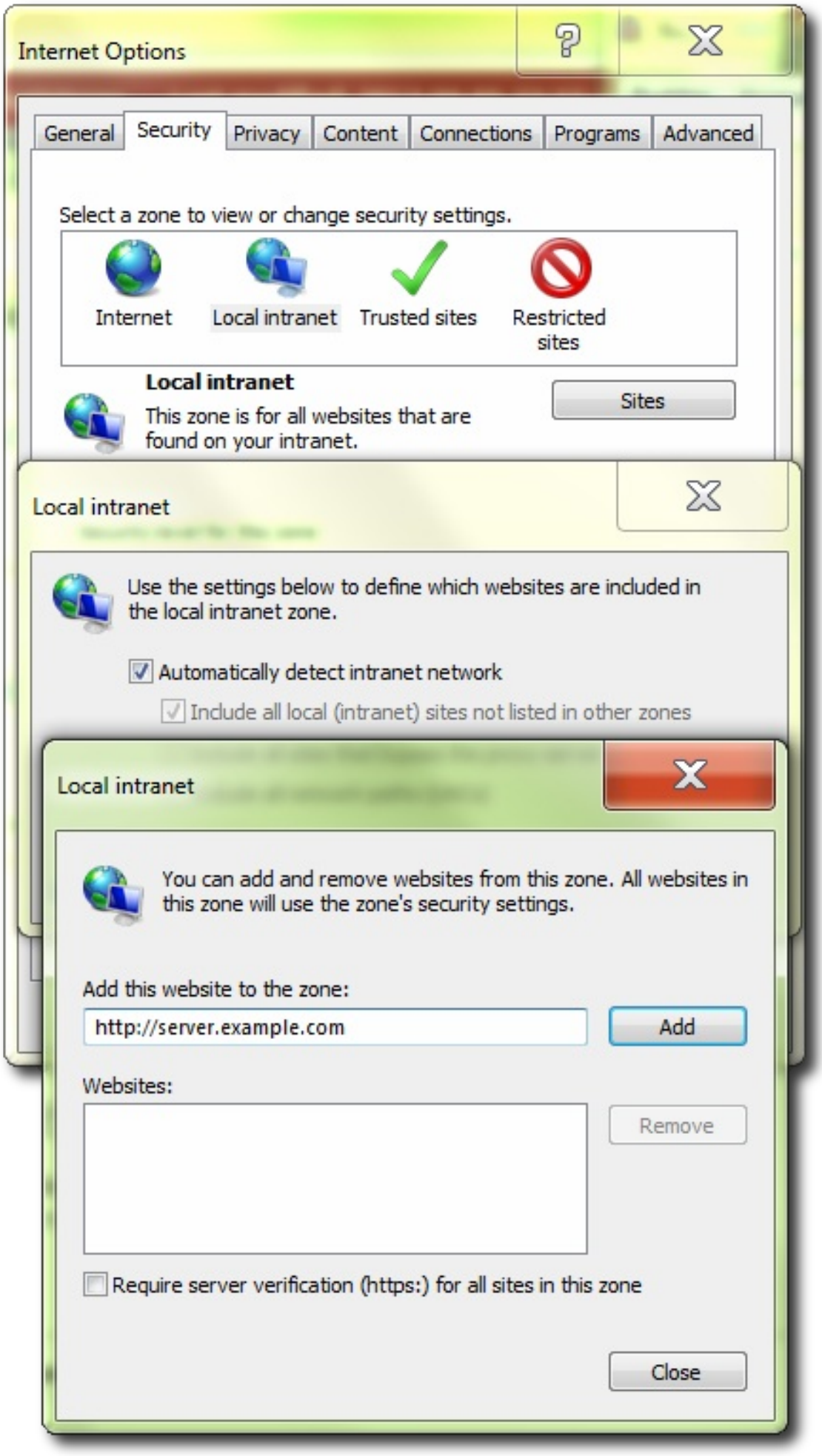
The recommended ZIP utility is WinZip. Examples in this guide usually use the Windows command prompt, so, optionally, install the WinZip CLI utility add-on. WinZip downloads are available from http://www.winzip.com.

Report a bug

## 5.8. Configuring Internet Explorer

Some Internet Explorer settings can prevent the JBoss ON login page from loading properly. By default, Internet Explorer is in *stealth* mode, which disables some JavaScript access for websites. To allow the login page to load, add the IP address of the JBoss ON server to the whitelist for Internet Explorer.

1. In Internet Explorer, click the gear icon in the upper right corner and select **Internet options**.
2. Open the **Security** tab, and select the **Local intranet** icon.
3. Click the **Sites** button.
4. Click the **Advanced** button at the bottom of the pop-up window.
5. Enter the JBoss ON server hostname or IP address in the **Add this webiste to the zone:** field, and click the **Add**.


Internet Options
General
Security
Privacy
Content
Connections
Programs
Advanced
Select a zone to view or change security settings.
Internet
Local intranet
Trusted sites
Restricted sites
Local intranet
This zone is for all websites that are found on your intranet.
Sites
Local intranet
Use the settings below to define which websites are included in the local intranet zone.
Automatically detect intranet network
Include all local (intranet) sites not listed in other zones
Local intranet
You can add and remove websites from this zone. All websites in this zone will use the zone's security settings.
Add this website to the zone:
http://server.example.com
Add
Websites:
Remove
Require server verification (https:) for all sites in this zone
Close

6. Close out the options windows.

Report a bug

## 5.9. Components Managed as Windows Services

On Windows, the **rhqctl.bat** script works analogously to **rhqctl** on UNIX-based platforms.

Components are installed and managed as Windows services. The Windows services run by default as the local system account (known as "Default" or ".\LocalSystem"). JBoss ON associates agent configuration preferences with users.

> **Note**
>
> When running **rhqctl.bat**, always start the command window with "Run As Administrator" to ensure the script can manipulate Windows services.

Make sure that custom agent preferences are correctly applied to the user running the RHQ Agent service.

There are a number of ways to achieve this:

**Install using RHQ_AGENT_RUN_AS_ME, and use --agent-preference**

This will run the service as the same user executing **rhqctl**, so the command line **--agent-preference** settings apply.

**Install using --agent-config <custom-agent-configuration.xml>**

Without ***RHQ_AGENT_RUN_AS_ME*** the user executing **rhqctl** will be different than the user running the service.

This option applies the custom configuration file on the RHQ Agent initial start, which contains the required settings to ensure the services run under the correct user.

**Update the RHQ Agent configuration post-install**

After installation the agent config can be updated using the agent prompt commands.

Without ***RHQ_AGENT_RUN_AS_ME*** the user executing **rhqctl** will be different than the user running the service. Without **--agent-config** the installed agent will have default configuration.

**Procedure 2.1. Setting the Required User Account for Windows Services**

1. Open **rhq-server-env.bat**.

2. Define the environment variables.

```
RHQ_SERVER_RUN_AS=.\username
RHQ_SERVER_PASSWORD=password

RHQ_STORAGE_RUN_AS=.\username
RHQ_STORAGE_PASSWORD=password

RHQ_AGENT_RUN_AS=.\username
RHQ_AGENT_PASSWORD=password
```

   The **RHQ_*_RUN_AS** parameters set the user account to use. The **RHQ_*_RUN_AS_ME** parameter uses the logged in user as the service account. If both parameters are set, then the **RHQ_*_RUN_AS_ME** parameter is the one which is used.

3. Explicitly grant the *log on as service* permission for the specified user account.

Report a bug

## 5.10. Planning the Storage Nodes

There must be at least one back-end storage database to store metrics data. This storage node is installed using the **rhqctl** script (the same as the server). The metrics storage database works as a cluster, so it is possible to have multiple nodes and to add and remove nodes as necessary.

There are some guidelines for planning the storage nodes:

- The storage node and server do not have to be located on the same machine.
- Installing a storage node does not require installing a server.
- At least one storage node must be installed before the server. (If the **rhqctl** script is run with just the **install** command, then it automatically installs a storage node first, then the server, then the agent.)
- Multiple storage nodes can be installed before installing the server. There are several benefits to installing multiple nodes:
  - For upgraded environments, it speeds data migration.
  - It can minimize the administrative and resource requirements that are incurred by deploying nodes after the server is running.

Report a bug

# 6. About the rhqctl Script

JBoss Operations Network has a control script which is used for basic lifecycle management for the server and storage nodes. It can open a server console and start and stop the server.

The control script (**rhqctl**) has two subcommands which are relevant to the installation process: **install** and **upgrade**.

Report a bug

## 6.1. Using the rhqctl Script

The **rhqctl** script has subcommands and options:

```
rhqctl [command] [[options]
```

For the installation process, the only relevant command is **install**.

There are a number of options with the **install** command which allow for more custom ways of configuring the JBoss ON server, depending on your needs.

**Example 2.1. Installing with No Options**

The simplest way to configure the server is to run the **install** command alone.

```
jsmith@server bin]$ ./rhqctl install
06:21:40,773 INFO  [org.jboss.modules] JBoss Modules version
1.3.3.Final-redhat-1

The [rhq.autoinstall.server.admin.password] property is required
but not set in [rhq-server.properties].
Do you want to set [rhq.autoinstall.server.admin.password] value
now?
yes|no: yes
rhq.autoinstall.server.admin.password (enter as plain text):
Confirm:
rhq.autoinstall.server.admin.password (enter as plain text):

The [jboss.bind.address] property is required but not set in [rhq-
server.properties].
Do you want to set [jboss.bind.address] value now?
yes|no: yes
jboss.bind.address: 0.0.0.0
Is [0.0.0.0] correct?
yes|no: yes
```

This installs all three management components:

- The server
- The storage database node
- The local agent

When the configuration process is complete, the server, storage node, and agent are not running, so these processes must be started manually.

```
[jsmith@server bin]# ./rhqctl start
```

**Example 2.2. Installing and Starting Services**

The **--start** option starts all services as soon as the installation process is complete. This is the same as running the **start** command immediately.

```
[jsmith@server bin]# ./rhqctl install --start
```

**Example 2.3. Installing Specific Services**

The **install** command configures the JBoss ON server, storage node, and agent all at the same time.

While it is recommended that all three management services be run on the same system (and from the same parent directory), there may be some environments where it is beneficial to run the JBoss ON server on a separate machine from the storage node. In other cases, it may be required to install the different services at different times.

The **install** command has options for each service. If that option is used, the only that service is installed; the other services are excluded.

For example, this installs the server, storage node, and agent in three separate command invocations:

```
[jsmith@server bin]# ./rhqctl install --storage --start
[jsmith@server bin]# ./rhqctl install --server --start
[jsmith@server bin]# ./rhqctl install --agent --start
```

If the services will be installed on the same system but separately, install the storage node first. The storage node needs to be installed and running when the server is installed.

**Table 2.4. Options for Installing JBoss ON**

| Option | Description |
|---|---|
| --start | Starts all services as soon as the installation process is complete. |
| --server | Installs the server. The server is installed by default; if this is specified, then the server is installed and other components are not installed (unless they are explicitly mentioned). |
| --storage | Installs the storage database node. The storage database node is installed by default; if this is specified, then the storage database and a companion agent are installed, but the server is not. |
| --storage-data-root-dir *directory* | Changes the directory where the storage data are stored. By default, the storage node directory is ***serverRoot*/jon-server-3.3.2.GA/rhq-data/**. |
| --agent | Installs the agent. The agent is installed by default; if this is specified, then the agent is installed and other components are not installed (unless they are explicitly mentioned). |

Report a bug

## 6.2. Attributes in the Properties File

All of the configuration for the JBoss ON server is pulled, at configuration time, from its **rhq-server.properties** file. Most of the configuration is defined by default:

- Database connection information
- The username and password for the database user
- The JBoss ON server port numbers
- The name for the server instance in the JBoss ON cloud
- The way to handle any existing schema in the JBoss ON database
- Server/agent communication settings, including SSL settings
- Connection and concurrency limits for the server

There are other settings, as well, but those are the most common ones. The attribute names and descriptions are listed below. Any of these settings can be edited before the **rhqctl** script is run to set new values. If no changes are made, there are three notable configuration areas:

- The default database configuration uses a PostgreSQL database installed on the same host as the JBoss ON server.
- The bind address (IP address) for the server is left blank, and the control script prompts for a value.
- The server name is left blank, and the default value is the server's hostname.

#### rhq-server.properties Attributes for Server Configuration

**rhq.server.high-availability.name**

Sets an optional name to use to identify the server within the JBoss ON server cloud. If this is not given, then the default value is the server hostname.

**jboss.bind.address**

Gives the IP address to use to connect to the JBoss ON server. If the server is available over all interfaces, then set this to **0.0.0.0**.

**rhq.autoinstall.database**

Sets how to handle any existing data in the JBoss ON database. The default is **auto**, which means that the installation process adds new schema but preserves any existing data. The other option is **overwrite**, which updates the schema and removes any existing data.

**rhq.autoinstall.server.admin.password**

Stores the encrypted server password, generated as a step of the **rhqctl install** command for new installations. The encrypted value can be manually generated using the **rhq-encode-value.(bat|sh)** script, and the value updated manually by the user.

**rhq.server.startup.web.http.port , rhq.server.startup.web.https.port**

Set the standard (HTTP) and secure (HTTPS) ports for the JBoss ON server. The default values are 7080 and 7443, respectively.

**rhq.server.database.type-mapping**

Gives the type or vendor of the database that is used by the JBoss ON server. This is either PostgreSQL or Oracle10g (Oracle10g is used for Oracle database versions 10, 11, and 12).

**rhq.server.database.connection-url**

The JDBC URL that the JBoss ON server uses when connecting to the database. This has the format (roughly) of *jdbc:db-type:hostname:port[:|/]db-name*.

An example is *jdbc:postgresql://localhost:5432/rhq* or *jdbc:oracle:oci:@localhost:1521:orcl*.

**rhq.server.database.user-name**

The name of the user that the JBoss ON server uses when logging into the database. The default is **rhqadmin**.

**rhq.server.database.password**

The password of the database user that is used by the JBoss ON server when logging into the database.

This password is stored in a hash. The default password is **rhqadmin**.

If a different password was created for the database user, encrypt the password using the ***serverRoot*/jon-server-3.3.2.GA/bin/rhq-encode-value.sh** script, and update the encrypted password value in the **rhq.server.database.password** attribute.

> **Note**
>
> The **rhq-encode-password.sh** script is deprecated, but remains in JBoss ON for backwards compatibility. It is recommended to discontinue use of the old script and begin using ***serverRoot*/jon-server-3.3.2.GA/bin/rhq-encode-value.sh** for password encryption.

**rhq.server.database.server-name**

The server name where the database is found. This must match the server in the connection URL. This is currently only used when connecting to PostgreSQL.

**rhq.server.database.port**

The port on which the database is listening. This must match the port in the connection URL. This is currently only used when connecting to PostgreSQL.

**rhq.server.database.db-name**

The name of the database. This must match the name found in the connection URL. This is currently only used when connecting to PostgreSQL.

**rhq.server.quartz.driverDelegateClass**

The Quartz driver used for connections between the server and the database. The

The Quartz driver used for connections between the server and the database. The value of this is set by the installer and depends on the type of database used to store the JBoss ON information. For PostgreSQL, this is **org.quartz.impl.jdbcjobstore.PostgreSQLDelegate**, and for Oracle, this is **org.quartz.impl.jdbcjobstore.oracle.OracleDelegate**.

Report a bug

## 7. Basic Setup: Installing the Server on Linux

**Procedure 2.2.**

1. Download the JBoss ON binaries from the Customer Support Portal.

    a. In the Customer Support Portal, click the **Downloads** tab, and then the **Downloads** icon in the page.

    b. Select the **JBoss Operations Network** link under the **System Management** area in the **Downloads** page.

    c. Download the **JBoss Operations Network 3.3 Base Distribution** package by clicking the **Download** icon.

    d. There are additional plug-in packs available for EAP, EDS, EWS, and SOA-P. If any of those plug-ins will be used with the JBoss ON server, then download them as well.

2. Unzip the server distribution to the desired home directory for JBoss ON. For example:

    ```
    [jsmith@server ~]# unzip jon-server-3.3.2.GA.zip -d /opt/jon
    ```

    This creates a version-specific installation directory, **/opt/jon/jon-server-3.3.2.GA**. A directory with this name should not exist prior to the unzip operation.

3. *Optional*. By default, the script assumes that the backend database is a PostgreSQL server running on the same system as the server. Other settings — such as the database password, the server port numbers, the server name, and the way it handles database schema — use predefined defaults. One parameter, the bind address for the server, is empty and prompted by the control script.

    To change any of these defaults or to set additional information, edit the **rhq-server.properties** file. This is briefly covered in Section 6.2, “Attributes in the Properties File”.

4. Run the JBoss ON control script to configure the server and other services. If the **rhq-server.properties** file is not edited, then the script prompts for a bind address for the server; this can be set to **0.0.0.0** and for rhq.autoinstall.server.admin.password.

```
[jsmith@server ~]# serverRoot/jon-server-3.3.2.GA/bin/rhqctl
install --start
06:21:40,773 INFO  [org.jboss.modules] JBoss Modules version
1.3.3.Final-redhat-1

The [rhq.autoinstall.server.admin.password] property is
required but not set in [rhq-server.properties].
Do you want to set [rhq.autoinstall.server.admin.password]
value now?
yes|no: yes
rhq.autoinstall.server.admin.password (enter as plain text):
Confirm:
rhq.autoinstall.server.admin.password (enter as plain text):

The [jboss.bind.address] property is required but not set in
[rhq-server.properties].
Do you want to set [jboss.bind.address] value now?
yes|no: yes
jboss.bind.address: 0.0.0.0
Is [0.0.0.0] correct?
yes|no: yes
```

This command does two things:

- It configures the JBoss ON server, a storage node, and an agent.
- It starts all services when the configuration process is complete.

5. Edit the **rhqctl** control script so that it runs as a system user rather than a **root** user.

```
[jsmith@server ~]# vim serverRoot/jon-server-
3.3.2.GA/bin/rhqctl
```

By default, the JBoss ON server runs as whatever user invokes the control script, but for security reasons, it is not recommended that the JBoss ON server process run as **root**.

> **Note**
>
> The user should be a system user who is not able to log into the system. When creating the user, set the **nologin** option to limit the level of access for the user.

In the **rhqctl** script, below the # comment lines at the top of the file, add a line to set a non-root user to run the JBoss ON server process. For example, this uses a system user named **jboss**.

```
...k
  #processname: standalone.sh

  su - jboss -s /bin/bash -c "/etc/init.d/rhqctl $*" &
```

6. It may take several minutes for the server process to start fully. Afterward, log into the server web UI to begin configuring resources.

   The server URL is *http://hostname:7080*. For example:

   ```
   http://server.example.com:7080
   ```

   The username is **rhqadmin** and the password is the value set when running the installer, or configured manually as the value of the rhq.autoinstall.server.admin.password.

Report a bug

## 8. Basic Setup: Installing the Server on Windows

**Procedure 2.3.**

1. When opening the command prompt, right-click the name or icon, and select **Run as Administrator**.

2. Download the JBoss ON binaries from the Customer Support Portal.

   a. In the Customer Support Portal, click the **Downloads** tab, and then the **Downloads** icon in the page.

   b. Select the **JBoss Operations Network** link under the **System Management** area in the **Downloads** page.

   c. Download the **JBoss Operations Network 3.3 Base Distribution** package by clicking the **Download** icon.

   d. There are additional plug-in packs available for EAP, EDS, EWS, and SOA-P. If any of those plug-ins will be used with the JBoss ON server, then download them as well.

3. Create a directory for the server to be installed in.

   Use a relatively short name. Path names longer than 19 characters can cause problems running the server or executing some tasks.

4. Unzip the server distribution to the desired home directory for JBoss ON. For example:

```
C:> winzip32 -e jon-server-3.3.2.GA.zip C:\jon
```

This creates a version-specific installation directory, **C:\jon\jon-server-3.3.2.GA**. A directory with this name should not exist prior to the unzip operation.

> **Important**
>
> Be careful when using the *extract all* command. Expanding the archive automatically creates a directory called *jon-server-VER.RELEASE/*, which is about 20 characters. Using *extract all*, instead of specifying the directory to which to extract the archive, can double the directory name by extracting to the archive name and then to a subdirectory — for example, **C:\example\jon-server-3.3.2.GA\jon-server-3.3.2.GA**. Using other tools may install it to a downloads directory such as **C:\Users\Administrator\Downloads**.
>
> If directory paths are too long, then installations on Windows can fail.
>
> It is recommended that you extract the archive to **C:\jon**, such as **C:\jon\jon-server-3.3.2.GA**.

5. Set the directory path to the JDK installation. For example:

```
set RHQ_JAVA_HOME=C:\Program Files\Java\jdk1.6.0_29
```

6. *Optional.* By default, the script assumes that the database is a PostgreSQL server running on the same system as the server. Other settings — such as the database password, the server port numbers, the server name, and the way it handles database schema — use predefined defaults. One parameter, the bind address for the server, is empty and prompted by the control script.

   To change any of these defaults or to set additional information, edit the **rhq-server.properties** file. This is briefly covered in Section 6.2, “Attributes in the Properties File”.

7. *Optional.* The Windows services run by default as the local system account (*Default* or *.\LocalSystem*). It is possible to configure the services to run as different users by setting the appropriate properties in the **rhq-server-env.bat** script.

   The **RHQ_*_RUN_AS** parameter sets the user account to use. The **RHQ_*_RUN_AS_ME** parameter uses the logged in user as the service account. If both parameters are set, then the **RHQ_*_RUN_AS_ME** parameter is the one which is used.

```
RHQ_SERVER_RUN_AS=.\username
RHQ_SERVER_PASSWORD=password
```

```
RHQ_STORAGE_RUN_AS=.\username
RHQ_STORAGE_PASSWORD=password

RHQ_AGENT_RUN_AS=.\username
RHQ_AGENT_PASSWORD=password
```

The defined user account must have the *log on as service* permission. This may need to be granted explicitly.

8. Run the JBoss ON control script to configure the server and other services. If the **rhq-server.properties** file is not edited, then the script prompts for a bind address for the server; this can be set to **0.0.0.0** and for rhq.autoinstall.server.admin.password.

```
C:\jon\jon-server-3.3.2.GA\bin> serverRoot/jon-server-
3.3.2.GA/bin/rhqctl install --start
06:21:40,773 INFO  [org.jboss.modules] JBoss Modules version
1.3.3.Final-redhat-1

The [rhq.autoinstall.server.admin.password] property is
required but not set in [rhq-server.properties].
Do you want to set [rhq.autoinstall.server.admin.password]
value now?
yes|no: yes
rhq.autoinstall.server.admin.password (enter as plain text):
Confirm:
rhq.autoinstall.server.admin.password (enter as plain text):

The [jboss.bind.address] property is required but not set in
[rhq-server.properties].
Do you want to set [jboss.bind.address] value now?
yes|no: yes
jboss.bind.address: 0.0.0.0
Is [0.0.0.0] correct?
yes|no: yes
```

This command does two things:

- Configure the JBoss ON server, a storage node, and an agent.
- Start all services when the configuration process is complete.

9. It may take several minutes for the server process to start fully. Afterward, log into the server web UI to begin configuring resources.

The server URL is *http://hostname:7080*. For example:

```
http://server.example.com:7080
```

The username is **rhqadmin** and the password is the value set when running the installer, or configured manually as the value of the rhq.autoinstall.server.admin.password.

Report a bug

## 9. Installing Additional Servers for High Availability

JBoss ON can be configured to run in a high availability cloud by configuring multiple server instances which all use the same SQL database backend. Because all of the servers share a backend, they all have the same set of data and inventory to use and all communicate with the same agents.

### Installing an Additional Server with an Agent and Storage Node

At a minimum, additional servers must be installed with the same SQL database information as the first JBoss ON server instance. The **rhq-server.properties** file must be edited to use the same database configuration as the original instance; the database properties are listed in Section 6.2, "Attributes in the Properties File". After editing the properties file for the database settings, the server can be installed as normal:

```
[jsmith@server ~]# serverRoot/jon-server-3.3.2.GA/bin/rhqctl install
--start
```

### Installing an Additional Server with a Separate Storage Node

With the default **install** command, a server, agent, and storage node are installed. In some high availability deployments, a storage node may not be installed with every server. In that case, the configuration for the existing storage node must be added to the server configuration as part of its installation process.

1. On the original server machine, check the **Administration >Storage Nodes** area for the list of IP addresses or hostnames for the storage nodes and for the client and gossip ports used by the nodes.

2. On the new server machine, before installing the server, edit the **rhq-server.properties** file to include the connection information for the storage nodes.

   Add each storage node in a comma-separated listed to the ***rhq.storage.nodes*** parameter. Then, add the client and gossip port values.

   ```
   [jsmith@server ~]# vim serverRoot/jon-server-3.3.2.GA/bin/rhq-
   server.properties

   rhq.storage.nodes=192.168.0.0,192.168.0.1,192.168.0.2
   rhq.storage.cql-port=9142
   rhq.storage.gossip-port=7100
   ```

3. Install the server and an agent. Specifying the **--server** and **--agent** options only installs those two components; the storage database is excluded.

```
[jsmith@server ~]# serverRoot/jon-server-3.3.2.GA/bin/rhqctl
install --server --agent --start
```

Report a bug

## 10. Installing Storage Nodes Before Installing the Server

It is possible to create multiple storage nodes before installing a server, and then install the server with those pre-installed nodes. This is also useful if the storage database will be on a separate, dedicated machine.

**Warning**

This is an advanced configuration. If the storage node or nodes within the cluster are not properly configured, then the cluster may not properly function.

**Warning**

Deploying a node lists that node's host in the cluster configuration and *any* allowed host can gain access to the data in the storage cluster.

Restrict access to the **rhq-storage-auth.conf** file so that the allowed hosts list cannot be altered to allow an attacker to gain access to the cluster and the stored data.

**Important**

Every storage node must use the same client (CQL) and gossip ports.

Additionally, the hostname and IP address of every storage node system must be fully resolvable in DNS or must be configured on each system's **hosts** file.

1. Determine the node and cluster configuration information to use.
   - Identify the hostname or IP address of each system which will host a node.
   - Define the two ports which the cluster uses for communication (9142 and 7100 by default).
2. *Before installing any storage node*, edit the storage properties file with all of the node and cluster information.

```
[jsmith@server ~]# vim serverRoot/jon-server-3.3.2.GA/bin/rhq-
storage.properties
```

-

For example, this configures three nodes, set in the ***rhq.storage.seeds*** parameter.

```
rhq.storage.cql-port=9142
rhq.storage.gossip-port=7100
rhq.storage.seeds=192.168.0.0, 192.168.0.1, 192.168.0.2
start=false
```

3. Install the storage node on each system, with its companion agent. This requires the IP address of the JBoss ON server, *even though the server is not yet installed*.

**Important**

Do not start the storage node or the agent at this point. Do not use the **--start** option with the installation script.

```
[jsmith@server ~]# serverRoot/jon-server-3.3.2.GA/bin/rhqctl
install --storage --agent-preference="rhq.agent.server.bind-
address=192.168.0.2"
```

**Note**

For Windows users, see Section 5.9, “Components Managed as Windows Services” for specific requirements for using **--agent-preference**.

4. For each storage node, edit its local **rhq-storage-auth.conf** file. This lists the hostnames or IP addresses for all of the storage nodes in the cluster, one per line.

```
[jsmith@server ~]# vim serverRoot/jon-server-3.3.2.GA/rhq-
storage/conf/rhq-storage-auth.conf

192.168.0.0
192.168.0.1
192.168.0.2
```

After the server is configured, the local agent will update the **rhq-storage-auth.conf** file with node hostnames or IP addresses as nodes are deployed and removed from the cluster.

5. Start each node.

```
[jsmith@server ~]# serverRoot/jon-server-3.3.2.GA/bin/rhqctl
start --storage
```

6. Before installing the server, edit the **rhq-server.properties** file to include the connection information for the storage nodes.

Add each storage node in a comma-separated listed to the ***rhq.storage.nodes*** parameter. Then, add the client and gossip port values.

```
[jsmith@server ~]# vim serverRoot/jon-server-3.3.2.GA/bin/rhq-server.properties

rhq.storage.nodes=192.168.0.0,192.168.0.1,192.168.0.2
rhq.storage.cql-port=9142
rhq.storage.gossip-port=7100
```

7. Specify the database settings in the **rhq-server.properties** file.

   The following database properties must be set to have the storage node properly register with the JBoss ON database.

   - rhq.server.database.connection-url
   - rhq.server.database.user-name
   - rhq.server.database.password

   These are the same settings used when installing an RHQ Server. See Section 6.2, "Attributes in the Properties File" for a description of these parameters.

8. Install the server and an agent. Specifying the **--server** and **--agent** options only installs those two components; the storage database is excluded.

   ```
   [jsmith@server ~]# serverRoot/jon-server-3.3.2.GA/bin/rhqctl install --server --agent --start
   ```

   If you are upgrading an existing JBoss ON agent, then run the upgrade script with the **--use-remote-storage-node** option, to load the storage database information from the properties file rather than installing a storage node.

   ```
   [jsmith@server]# serverRoot/jon-server-3.3.2.GA/bin/rhqctl upgrade --use-remote-storage-node=true
   ```

Report a bug

# 11. Managing the Server Service

## 11.1. Starting the Server and Other Services

The simplest way to start all installed services (server, agent, and storage node) is simply to run the script with the **start** command.

```
[jsmith@server ~]$ serverRoot/bin/rhqctl start
Trying to start the RHQ Server...
RHQ Server (pid 27547) is starting
```

Any individual service can be started using the appropriate option (**--server**, **--storage**, or **--agent**).

```
[jsmith@server ~]$ serverRoot/bin/rhqctl start --server
```

The **RHQ_JAVA_HOME** environment variable must be set on Red Hat Enterprise Linux systems for the server to start. This can be set to a general value like **/usr/**.

**Note**

The server must be started using the **rhqctl** script, not the **rhq-server.sh** script.

Any agents installed with the server must be started using the **rhqctl** command. It must not be started using the **rhq-agent.sh** script.

Additionally, the agent must be started without requiring any user intervention. The **RHQ_AGENT_PASSWORD_PROMPT** parameter should always be commented out or set to false so that no password is required to start the agent.

Report a bug

## 11.2. Opening the Server in a Console

When the server is running as a service on either Windows or Linux, it is running in the background. It is possible to open the server in a console window, using the control script:

1. Stop the JBoss ON server.

```
[jsmith@server ~]$ serverRoot/jon-server-3.3.2.GA/bin/rhqctl
stop
```

2. Run the **rhqctl** script with the **console** command.

```
[jsmith@server ~]$ serverRoot/jon-server-3.3.2.GA/bin/rhqctl
console --server
```

Report a bug

# Chapter 3. Upgrading JBoss ON Servers and Storage Nodes

An upgrade procedure for JBoss Operations Network essentially overlays the new JBoss ON packages and libraries over the existing configuration and databases. The upgrade procedure, then, is very similar to the installation process. The new packages need to be installed, and then the server is configured through the same setup script. The difference is that the server reuses its existing databases and data so that the configuration from the previous installation is preserved.

Report a bug

## 1. Upgrade Notes

- **It is not possible to revert your JBoss ON server to the previous version after it is upgraded. Back up all data before upgrading.**
- There will be a minimal loss of monitoring data because of the downtime required when the server and agents are being upgraded. Additionally, any monitoring data for the JBoss ON server will be lost, if the server is included in the inventory.
- The JBoss ON servers must be upgraded before the JBoss ON agents can be upgraded.
- When an agent is upgraded, any files with the **`.sh`** or **`.bat`** extension found in the agent's **`/bin`** directory will be copied to the new agent's **`/bin`** directory.
- Upgrading the JBoss ON server essentially creates a new server instance that replaces the old instance. If the JBoss ON server was added to the inventory, then the old JBoss ON server resource must be deleted from the inventory because it will not be a usable resource after upgrade. Once the upgrade process is complete, then the JBoss ON server must be added to the inventory again and all of the previous configuration for that resource (like alerts, scheduled operations, and group membership) must be redone.
- *All* JBoss ON servers in the high availablity cloud must be stopped when one is upgraded. Otherwise, the installer will hang when it tries to contact the database and the database is unavailable because it is in use by another JBoss ON server.
- Do not copy the new server installation on top of a previous server installation.
- *On Windows.* When configuring JBoss ON servers as services on Windows, it was possible to set the **`RHQ_SERVER_RUN_AS`** parameter without setting a password. In JBoss ON 3.3, the **`RHQ_SERVER_PASSWORD`** parameter is required with the **`RHQ_SERVER_RUN_AS`** parameter.
- *On Windows.* Users who wish to update the Java implementation from 32 bit to 64 bit on a Windows service running JBoss ON are required to uninstall and re-install that Windows service. For more details on this process, see steps 2 and 3 in Section 2, “Uninstalling the Server”.

Report a bug

## 2. About the Upgrade Script

As with installation (Section 6, "About the rhqctl Script"), the **rhqctl** script is used to manage server migrations. The upgrade command, much like the install command, handles all three management components on the server system:

- Upgrades the JBoss ON server.
- Upgrades the JBoss ON agent *on the same system as the JBoss ON server*. [1]
- Upgrades a JBoss ON storage node. If upgrading from a 3.1 (or older) deployment, the upgrade process installs a storage node.

The upgrade script requires the original location of the server and agent directories.

### Options for Upgrading JBoss ON

**--list-versions**

Prints the installed version of the servers and storage nodes in the topology. This must be run from a JBoss ON 3.3 server that is newly-installed or already upgraded. This cannot be run from a standalone storage node, only a server.

> **Note**
>
> When this option is used, any other options are ignored.

**--from-server-dir *directory***

Gives the directory path to the server to be upgraded.

**--from-agent-dir *directory***

Specifies the path, relative to the server installation directory in **--from-server-dir**s>. The default location is ***serverRoot*/jon-server-3.3.2.GA/rhq-agent/**.

> **Note**
>
> Only required when the agent is installed in a custom location.

**--storage-data-root-dir *directory***

Specifies the location to use for the storage node (by default, installed as part of upgrading) if it should use a non-default directory. This is useful if the default directory is not writable by the JBoss ON user, such as **`/var/lib`**.

> **Note**
>
> For upgrading 3.1 or older servers.

**`--storage-schema`**

Updates the storage cluster schema. This is done after all of the storage nodes and servers have been upgraded and are running. This must be run from a JBoss ON 3.3 server that is newly-installed or already upgraded. This cannot be run from a standalone storage node, only a server.

The running time for the schema update varies depending on the schema changes being made and should not be interrupted.

> **Note**
>
> When this option is used, any other options are ignored.

**`--use-remote-storage-node [true | false]`**

Sets whether to use a local storage node or one on another system. By default, storage nodes and servers are on the same system; if this option is set to **`true`**, then the server configuration is used to identify the remote storage node connection information.

The running time for the schema update varies depending on the schema changes being made and should not be interrupted.

> **Note**
>
> When this option is used, any other options are ignored.

Report a bug

## 3. Upgrading a 3.1.x Server and Server Plug-ins

JBoss ON 3.1.2 and later can be upgraded to JBoss ON 3.3. Earlier JBoss ON 3.1 servers must first be upgraded to 3.1.2, and they then can be upgraded to JBoss ON 3.3.

Not every step in this upgrade procedure applies to every JBoss Operations Network installation. Run through the steps in order, and perform the ones necessary for your deployment.

**Warning**

It is not possible to revert your JBoss ON server to the previous version after it is upgraded. Back up all data before upgrading.

**Note**

To make the migration process go faster, deploy multiple storage nodes before upgrading the server. This is covered in Section 10, “Installing Storage Nodes Before Installing the Server”.

1. For older 3.1 versions, upgrade to JBoss ON 3.1.2 or the latest release.

2. Stop the JBoss ON agent running on the *server* machine. If the agent is running as a service, then stop the system service. It is also possible to stop it at the command prompt:

```
[jsmith@server ~]$ agentRoot/rhq-agent/bin/rhq-agent.sh
> exit
```

3. *Windows only.*. If the **RHQ_AGENT_RUN_AS** or **RHQ_AGENT_RUN_AS_ME** parameter was set in the agent's **rhq-agent-env.bat** file, then there must be a password, and the password prompt must be disabled.

```
RHQ_AGENT_PASSWORD=secret
RHQ_AGENT_PASSWORD_PROMPT=false
```

**Note**

If one of the **RHQ_AGENT_RUN_AS*** parameters is set without a password, then the agent upgrade process hangs.

Alternatively, the **RHQ_AGENT_RUN_AS*** parameter can be removed prior to upgrading.

4. Clean up the JBoss ON configuration. It is easier to clean up the configuration before migration than it is after.

   - Remove any unused or out of service platforms from the inventory.

   - If the older JBoss ON server was added to the JBoss ON inventory, then remove it.

     The old JBoss ON server must be removed from the inventory because it is no longer a usable resource.

5. Stop all servers. For example:

```
[jsmith@server ~]$ serverRoot/jon-server-3.1.2.GA/rhq-
server.sh stop
```

**Important**

If the upgraded JBoss ON server will use a database that existing JBoss ON instances are also using, then all of the existing JBoss ON instances have to be stopped. Otherwise, the installer will hang when it tries to contact the database and the database is unavailable because it is in use by another JBoss ON server.

6. *Windows only.* If the server is running as a service, uninstall that service.

```
C:> cd C:\jon\jon-server-3.1.2\bin
C:\jon\jon-server-3.1.2\bin> ./rhq-server.bat remove
```

7. Back up the server database before going through the upgrade script.

   In case there is a problem with the upgrade process, the backup allows you to restore to its previous state.

8. If the **rhq-server.sh** or **rhq-server-wrapper.conf** files have been customized, back up those files. Changes made to these files must be reapplied manually after the upgrade script is run.

9. Unzip the server packages.

```
[jsmith@server ~]$ unzip jon-server-3.3.2.GA.zip -d
serverRoot/jon-server-3.3.2.GA
```

   **Important**

   Do not copy the new server installation on top of a previous server installation.

   Do not delete the existing JBoss ON installation directory, since it is used during the upgrade. **rhqctl upgrade** merges the old **rhq-server.properties** file into the new **rhq-server.properties** file.

   The directory structure within the server package gives the new server installation directory a version-specific name, such as **/opt/jon/jon-server-3.3.2.GA**.

10. Run the **upgrade** command.

    There are three critical options that can be used with the **upgrade** command:

- One option is always required: **--from-server-dir**, which identifies the original server's installation directory.

- If there is a local agent, then the **--from-agent-dir** is also required. If there is no agent, one will be installed when installing the storage node. By default in 3.3, this is installed within the same parent directory as the server's installation directory (such as **/opt/jon**).

- Decide where to host the new storage node. The **upgrade** command will create a new local storage node by default. Alternatively, a storage node can be created first; the storage node configuration is then added to the properties file and signaled with the **--use-remote-storage-node** option.

For example, this runs the upgrade for a local server and agent and creates a new local storage node:

```
[jsmith@server ~]$ ./serverRoot/jon-server-3.3.2.GA/bin/rhqctl
upgrade --from-server-dir /opt/rhq/rhq-server-old --from-
agent-dir /home/rhq/rhq-agent-old
```

To use a remote storage node:

a. On a different system, create the new storage node, as in Section 10, “Installing Storage Nodes Before Installing the Server”.

b. Edit the *new* **rhq-server.properties** file to point to the new storage node.

```
rhq.storage.cql-port=9142
rhq.storage.gossip-port=7100
rhq.storage.seeds=192.168.0.0, 192.168.0.1, 192.168.0.2
start=false
```

c. Run the upgrade script with the **--use-remote-storage-node** option.

```
[jsmith@server ~]$ ./serverRoot/jon-server-
3.3.2.GA/bin/rhqctl upgrade --from-server-dir
/opt/rhq/rhq-server-old --from-agent-dir /home/rhq/rhq-
agent-old --use-remote-storage-node
```

11. After upgrading to JBoss ON 3.3, to alter any defaults for the storage node, create a ***serverRoot*/jon-server-3.3.2.GA/bin/rhq-storage.properties**. This file can be used to set any **--storage-config** options. These include the directories for data storage, host and port information, and several other options.

12. Upgrade the storage cluster schema.

a. Start all storage nodes. **Do not start any servers or agents.**

```
[jsmith@server ~]$ serverRoot/jon-server-
3.3.2.GA/bin/rhqctl start --storage
```

b. *On a JBoss ON server system*, re-run the **upgrade** command with the **--storage-schema** option. The command only has to be run once for the storage schema changes to be propagated to the storage cluster.

```
[jsmith@server ~]$ serverRoot/jon-server-
3.3.2.GA/bin/rhqctl upgrade --storage-schema
```

13. *Important.* Migrate the historical monitoring data.

There is a command-line script available to migrate all existing monitoring data. In most cases, this should be run at the same time the server is migrated.

> **Note**
>
> For large databases, it can take hours to migrate monitoring data, and the process **must not** be interrupted. Consider performing a migration during an extended period of low use.
>
> The data migrator tool can provide an estimate of how long the migration will take, to assist with planning.

If the data migrator tool is not run, all of the measurement data from the old server is no longer available. Also, if there is a large gap between when the server is upgraded and the data migration is run, any new monitoring data collected between the server upgrade and the data migration will be lost.

```
[jsmith@server ~]$ ./rhq-data-migration.sh
```

14. Check the **rhq-server.properties** file to make sure any edits were properly merged in. While this merge process should migrate all the values properly, it is still good practice to verify that the old properties file have been properly copied into the new properties file after the upgrade has completed.

15. If the **rhq-server.sh** or **rhq-server-wrapper.conf** files were customized, reapply any changes.

16. *Optional.* Additional plug-in packs for specific needs (such as supporting management tasks for other layered Red Hat JBoss Middleware products) are available for installation separate from the core JBoss ON packages.

Each plug-in pack has at least one (and sometimes more than one) agent plug-in. Each zip file for the plug-ins has a **README.txt** file with specific setup instructions.

> **Note**
>
> If there are multiple JBoss ON servers in a high availability setup, the agent plug-in pack only has to be installed once. The other servers will pick up the plug-ins as part of the high availability polls.

The plug-in files can be unzipped anywhere. For example:

```
[jsmith@server ~]$ unzip jon-plugin-pack-agent_plugin_name-3.3.2.GA.zip -d /opt/jon/jon-server-3.3.2.GA
```

17. Start the server, agent, and storage node.

```
[jsmith@server ~]$ serverRoot/jon-server-3.3.2.GA/bin/rhqctl start
```

18. *Optional.* Add the new JBoss ON server as a resource in the inventory.

Report a bug

## 4. Upgrading a 3.2 Server, Storage Nodes, and Server Plug-ins

Some steps in these upgrade procedures do not apply to all JBoss Operations Network installations. Run through each procedure and perform the steps necessary for your particular deployment.

> **Warning**
>
> It is not possible to revert your JBoss ON server to the previous version after it is upgraded. Back up all data before upgrading.

### Procedure 3.1. Prepare the Current Installation For Upgrade

1. Stop the JBoss ON agent running on the *server* machine.

```
[jsmith@server ~]# serverRoot/jon-server-3.2.0.GA/bin/rhqctl stop --agent
```

Agents installed with the server must be updated with the JBoss ON server. All other agents will update themselves automatically when the server is upgraded.

2. *Windows only.* If the **RHQ_AGENT_RUN_AS** or **RHQ_AGENT_RUN_AS_ME** parameter was set in the agent's **rhq-agent-env.bat** file, then there must be a password, and the password prompt must be disabled.

```
RHQ_AGENT_PASSWORD=secret
RHQ_AGENT_PASSWORD_PROMPT=false
```

**Important**

If one of the **RHQ_AGENT_RUN_AS*** parameters is set without a password, then the agent upgrade process hangs.

Alternatively, the **RHQ_AGENT_RUN_AS*** parameter can be removed prior to upgrading.

3. Clean up the JBoss ON inventory. It is easier to clean up the configuration before migration than it is after.

   - Remove any unused or out-of-service platforms from the inventory.
   - If the older JBoss ON server was added to the JBoss ON inventory, then remove it.

     The old JBoss ON server must be removed from the inventory because it is no longer a usable resource.

4. Stop all servers and storage nodes.

   Stop the JBoss ON server which is being upgraded *as well as* any currently running JBoss ON instances in the cloud.

   **Important**

   If the upgraded JBoss ON server will use a database that existing JBoss ON instances are also using, then all of the existing JBoss ON instances have to be stopped. Otherwise, the installer will hang when it tries to contact the database and the database is unavailable because it is in use by another JBoss ON server.

   ```
   [jsmith@server ~]# serverRoot/jon-server-3.2.0.GA/bin/rhqctl
   stop
   ```

5. Back up the server database before running the upgrade script.

   If a problem with the upgrade process occurs, the backup allows you to restore to its previous state.

### Procedure 3.2. Upgrade Each Standalone Storage Node

Upgrade each standalone storage node before upgrading the server.

1. Stop all 3.2 processes for the storage node and agent.

```
[jsmith@server ~]# serverRoot/jon-server-3.2.0.GA/bin/rhqctl
stop
```

2. Unzip the server packages.

```
[jsmith@server ~]# unzip jon-server-3.3.2.GA.zip -d
serverRoot/jon-server-3.3.2.GA
```

> **Important**
>
> Do not copy the new server installation on top of a previous server installation.
>
> Do not delete the existing JBoss ON installation directory, since it is used during the upgrade. **`rhqctl upgrade`** merges the old **`rhq-server.properties`** file into the new **`rhq-server.properties`** file.

The directory structure within the server package gives the new server installation directory a version-specific name, such as **`/opt/jon/jon-server-3.3.2.GA`**.

3. Verify all Database properties are correctly set in *serverRoot*/jon-server-3.2.0.GA/bin/rhq-server.properties. See Section 6.2, “Attributes in the Properties File” for the database properties required.

4. Run the upgrade script.

```
[jsmith@server ~]# serverRoot/jon-server-3.3.2.GA/bin/rhqctl
upgrade --from-server-dir serverRoot/jon-server-3.2.0.GA
```

> **Note**
>
> If a storage node is on the same system, those storage nodes will be upgraded with the **`upgrade`** is issued to the JBoss ON server. Standalone (storage nodes not co-located with a JBoss ON server) still require individual upgrades.

### Procedure 3.3. Upgrade Each Server

1. If the **`rhq-server.sh`** or **`rhq-server-wrapper.conf`** files have been customized, back up those files. Changes made to these files must be reapplied manually after the upgrade script is run.

2. Run the **`rhqctl`** script with the **`upgrade`** subcommand. For example:

```
[jsmith@server ~]# ./serverRoot/jon-server-3.3.2.GA/bin/rhqctl
upgrade --from-server-dir /opt/rhq/rhq-server-old --from-
agent-dir /home/rhq/rhq-agent-old
```

The **--from-server-dir** option is required to identify the server being migrated.

3. Check the upgraded **rhq-server.properties** file against the pre-upgrade server's copy to make sure any edits were properly merged in. It is good practice to verify that the old properties file has been properly copied into the new properties file at this step of the upgrade process.

4. If the **rhq-server.sh** or **rhq-server-wrapper.conf** files were customized, reapply any changes.

5. Download and expand Plug-in packs separately before proceeding.

   Plug-in packs for specific needs (such as supporting management tasks for other layered Red Hat JBoss Middleware products) are available for installation separate from the core JBoss ON packages.

   See Section 7, "Basic Setup: Installing the Server on Linux" or Section 8, "Basic Setup: Installing the Server on Windows" for instructions to download Agent Plug-in packs.

   Each plug-in pack has at least one (and sometimes more than one) agent plug-in. Each zip file for the plug-ins has a **README.txt** file with specific setup instructions.

   > **Note**
   >
   > If there are multiple JBoss ON servers in a high availability setup, the agent plug-in pack only has to be installed once. The other servers will pick up the plug-ins as part of the high availability polls.

   The plug-in files can be unzipped anywhere. For example:

   ```
   [jsmith@server ~]# unzip jon-plugin-pack-agent_plugin_name-3.3.2.GA.zip -d /opt/jon/jon-server-3.3.2.GA
   ```

**Procedure 3.4. Update the Storage Cluster Schema**

1. Start all storage nodes. **Do not** start any servers or agents.

   ```
   [jsmith@server ~]# serverRoot/jon-server-3.3.2.GA/bin/rhqctl start --storage
   ```

2. *On a JBoss ON server system*, re-run the **upgrade** command with the **--storage-schema** option. The command only has to be run once for the storage schema changes to be propagated to the storage cluster.

   ```
   [jsmith@server ~]# serverRoot/jon-server-3.3.2.GA/bin/rhqctl upgrade --storage-schema
   ```

**Procedure 3.5. Upgrade Each Agent Not Configured for Auto-Update**

1. Click Section 7, "Manually Upgrading the JBoss ON Agent" and follow the direction to upgrade each Agent.

2. Return to this procedure once you have completed the Agent Upgrade steps.

**Procedure 3.6. Start the Upgraded Installation**

1. Start the server, agent, and storage node.

```
[jsmith@server ~]# serverRoot/jon-server-3.3.2.GA/bin/rhqctl
start
```

2. *Optional.* Add the new JBoss ON server as a resource in the inventory.

Report a bug

## 5. Re-Installing the Server

When the JBoss ON server is initially configured, there is a set of flags in the **rhq-server.properties** which indicates that this is an initial setup instead of an upgrade (***rhq.autoinstaller.****). When the initial configuration is complete, is complete, the autoinstaller is disabled (so even setting the ***rhq.autoinstaller.**** properties in **rhq-server.properties** does not re-initiate the server configuration).

To re-install the server, remove the entire home directory for the server, and then unzip the original JBoss ON server archive and configure the server as if it were all new, as in Section 7, "Basic Setup: Installing the Server on Linux".

Report a bug

---

[1] Agents installed from a JAR file are upgraded automatically by the server. Agents installed through an RPM can be updated using the **rpm -Uvh** command with the new package.

# Chapter 4. Uninstalling the JBoss ON Server

Because both the JBoss Operations Network server and agent are extracted archive files, removing a server or agent ultimately consists of deleting those files.

Report a bug

## 1. Uninstalling an Agent on a Managed System

### 1.1. Removing an Agent on Linux (JAR)

> **Note**
>
> This procedure is for removing a standalone agent. If the agent was installed with the JBoss ON server or storage node, then use the **rhqctl** script to stop and remove it.

1. Stop the agent.
2. Delete the agent from JBoss ON topology.
    a. In the JBoss ON UI, click the **Administration** tab in the top menu.
    b. In the **Topology** section in the left menu, select the **Agents** item.
    c. Select the row for the agent to delete, in the list of installed agents.
    d. Click the **Delete** button at the bottom of the page.
    e. Confirm that the agent should be deleted.
3. On the managed system, delete the agent's installation directory.

Report a bug

### 1.2. Removing an Agent RPM

1. Stop the agent service.

```
[jsmith@server ~]# service jon-agent stop
```

2. Delete the agent from JBoss ON topology.
    a. In the JBoss ON UI, click the **Administration** tab in the top menu.

b. In the **Topology** section in the left menu, select the **Agents** item.

c. Select the row for the agent to delete, in the list of installed agents.

d. Click the **Delete** button at the bottom of the page.

e. Confirm that the agent should be deleted.

3. If the package was installed using **yum**, then use the **yum** to remove the package:

```
[jsmith@server ~]# yum remove jboss-on-agent jboss-on-agent-
init
```

If the RPM package was installed using **rpm**, then uninstall it using **rpm**:

```
[jsmith@server ~]# rpm -e jboss-on-agent-3.3.2.GA jboss-on-
agent-init-3.3.2.GA
```

Report a bug

## 1.3. Removing an Agent on Windows

**Note**

This procedure is for removing a standalone agent. If the agent was installed with the JBoss ON server or storage node, then use the **rhqctl** script to stop and remove it.

1. Stop the agent.

2. Delete the agent from JBoss ON topology.

a. In the JBoss ON UI, click the **Administration** tab in the top menu.

b. In the **Topology** section in the left menu, select the **Agents** item.

c. Select the row for the agent to delete, in the list of installed agents.

d. Click the **Delete** button at the bottom of the page.

e. Confirm that the agent should be deleted.

3. If the agent is configured as a Windows service, then remove it as a service.

```
> rhq-agent.bat remove
```

4. Delete the agent's installation directory.

Report a bug

## 2. Uninstalling the Server

Removing a server still leaves the database and its information intact, so historic data remain available directly from the database itself.

1. If this is the only JBoss ON server, then stop all agents. If there will be other JBoss ON servers in the topology, then agents managed by this server will naturally migrate over to the other servers in the high availability topology.

2. Stop the server.

```
> serverRoot/jon-server-3.3.2.GA/bin/rhqctl stop
```

3. If the server is configured as a Windows service, then remove it as a service.

```
> C:\rhq\jon-server-3.3.2.GA\bin\rhqctl.bat remove
```

4. Delete the server's installation directory.

Report a bug

# Chapter 5. Installing and Upgrading an Agent on a Managed Platform from the JAR File

JAR files to install the JBoss Operations Network agent on Red Hat Enterprise Linux, Windows, Solaris, AIX, and other *nix distributions are available as a download from the JBoss ON server.

> **Important**
>
> This is for installing an agent on a platform which will be managed by JBoss ON. If this system will host a JBoss ON server, then install the agent as part of the server installation process, as described in Chapter 2, *Installing the JBoss ON Server*.

Report a bug

## 1. Before Installing the Agent

### 1.1. Verify the Parent Directory Permissions

During the update process, files are written to the directory where the agent is currently installed. This means the parent directory of the agent's install directory must be writable by the user that is running the agent.

For example, if the agent's **rhq-agent-env.sh** file specifies ***$RHQ_AGENT_HOME*** as **/opt/rhq-agent-parent/rhq-agent**, the agent must have write permissions on the **/opt/rhq-agent-parent** directory.

Report a bug

### 1.2. Setting up the JRE for the JBoss ON Agent

The JBoss ON agent requires either Java 6 or Java 7 JRE.

1. Download and install the appropriate version of the JRE, if necessary.
2. Set the **RHQ_JAVA_HOME** environment variable to the installation directory.
    a. Open the **.bashrc** for the system user that will run JBoss ON. For example:

    ```
    vim /home/jon/.bashrc
    ```

    b. Add a line to set the **RHQ_JAVA_HOME** environment variable to the specific JRE directory. For example:

    ```
    export RHQ_JAVA_HOME=/usr/lib/jvm/jre-1.6.0-openjdk/
    ```

3. Set the system to use the correct version of the JRE using the system **alternatives** command. The selected version has the ***+** symbols by it.

```
/usr/sbin/alternatives --config java

There are 5 programs which provide 'java'.

  Selection    Command
-----------------------------------------------
   1           /usr/lib/jvm/jre-1.5.0-sun/bin/java
   2           /usr/lib/jvm/jre-1.4.2-gcj/bin/java
   3           /usr/lib/jvm/jre-1.6.0-sun/bin/java
*+ 4           /usr/lib/jvm/jre-1.6.0-openjdk/bin/java
   5           /usr/lib/jvm/jre-1.6.0-bea/bin/java

Enter to keep the current selection[+], or type selection
number:
```

Report a bug

## 1.3. Configuring the Java Path

The agent requires that the path to the Java home directory is explicitly set as an environment variable.

Report a bug

## 1.4. Picking the Agent System User

Before installing the agent, plan what system user and group to use to run the agent. The given user can have an impact on how resources are discovered and how they should be configured for management.

The common types of servers which JBoss ON manages are:

- JBoss EAP servers
- PostgreSQL databases
- Tomcat servers
- Apache servers
- Generic JVMs

For the agent to be able to discover a resource requires, at a minimum, that the agent have read access to that resource's configuration. Some resource types may require more than just read access. For JBoss EAP resources, for example, the agent must have read

permissions to the **`run.jar`** file, plus execute and search permissions for every directory in the path to the **`run.jar`** file.

Read access or even root access may not be sufficient for some resource types. Tomcat servers can only be discovered if the JBoss ON agent and the Tomcat server are running as the same user. The same is true for JVMs and JMX servers with the attach API.

The system user which the agent runs as impacts several common agent tasks:

- Discovery
- Deploying applications
- Executing scripts
- Running start, stop, and restart operations
- Creating child resources through the JBoss ON UI
- Viewing and editing resource configuration

There is a general assumption that the agent runs as the same user as the managed resources, and this is the easiest option to manage resources effectively.

> **Important**
>
> While it is possible to run the JBoss ON agent as the root user, and in some limited contexts that may be the simple choice, consider the security implications of running a service as root before setting up the agent.
>
> Generally, services should be run with the *least amount of access required to perform their operations*. This is because if a service is ever compromised, its access permissions can be exploited by an attacker.
>
> The Red Hat Enterprise Linux *Security Guide* contains a section on security guidelines and links to security planning documents. There are similar recommendations in the Windows documentation.

When the JBoss ON agent is installed from the agent installer JAR file, the system user and group who own the agent installation files is the same user who installs the JAR. So, a special system user can be created or selected, and then the agent can be installed by that user.

If the agent and the resource are run as different users and the agent needs to perform some actions as the resource user, there are a few configuration options, depending on what needs to be done:

- Configure scripts or operations to run using **sudo**. For long-running operations, such as starting a service or a process, the user which executes the script should be the same as the resource user because that user will have the proper authorization and permissions.
- Set start script environment variables to use the resource's principal and credentials, if available.
- *For JVM or JMX servers.* Select the connection configuration based on the user settings. For different users, use JMX remoting. For the same user, use either JMX remoting or the attach API.

**Table 5.1. Cheat Sheet for Agent and Resource Users**

| Resource | User Information |
|---|---|
| PostgreSQL | No effect for monitoring and discovery. The agent user must have read/write permissions to the PostgreSQL configuration file for configuration viewing and editing. |
| Apache | No effect for monitoring and discovery. The agent user must have read/write permissions to the Apache configuration file for configuration viewing and editing. |
| Tomcat | Must use the same user or it can not be discovered. |
| JMX server or JVM | Different users are fine when using JMX remoting; cannot be discovered with different users and the attach API |
| JBoss AS/EAP | Different users are all right, but requires read permissions on run.jar and execute and search permission on all ancestor directories for run.jar |

Report a bug

## 2. Installing the Agent from JAR File

1. Point your browser to the download URL on the server. For example:

   ```
   http://server.example.com:7080/agentupdate/download
   ```

   Save the agent binary update **.jar** in a directory where you want to install the agent. The file you save should have a **.jar** extension.

2. Copy the agent update binary **.jar** you downloaded from the JBoss ON server to the directory.

3. Install the JAR:

```
java -jar downloaded_agent_jar_file.jar --install
```

This will tell the agent update binary to extract the JBoss ON agent distribution and install a fresh copy of it in the **rhq-agent** subdirectory.

> **Important**
>
> Do not install the agent in a directory with spaces in the name, such as **C:\Program Files**.
>
> Installing the agent in a directory with spaces in the pathname can cause problems for the agent establishing a connection with certain types of resources, including some JBoss services.

4. Set the path to the JRE as an environment variable for the agent. The agent requires that the Java home directory is set explicitly in its configuration.

Open the ***agentRoot*/rhq-agent/bin/rhq-agent-env.sh** file, and uncomment or add the line for the **RHQ_JAVA_HOME** variable.

```
export RHQ_JAVA_HOME=/usr
```

5. Start the agent to launch the setup process.

```
agentRoot/rhq-agent/bin/rhq-agent.sh
```

> **Note**
>
> It is possible to skip the setup wizard by submitting the configuration all at once. Section 3, “Silently Installing an Agent” has the details for setting up an file that can pass the configuration directly to the agent installer.

6. As prompted, supply the information to configure the agent and the server connection.

```
[Agent Name] agentdomain.example.com
[Agent Hostname or IP Address] agentdomain.example.com
[Agent Port] 16163
[JON Server Hostname or IP Address] server.example.com
[JON Server Port] 7080
native enable
```

- The agent name must be unique among all agents in the JBoss ON deployment. By default, the name is the fully-qualified domain name of the host machine.

- The port is the one that the agent uses to listen for incoming messages from the

- The port is the one that the agent uses to listen for incoming messages from the server. This is ***rhq.agent.server.bind-port*** in the configuration file, if the default value isn't used.

- The server hostname and port are used by the agent to connect to a server to register itself with the JBoss ON system. This is not necessarily the primary server that the agent will use after registration. In the configuration file, these are ***rhq.agent.server.bind-address*** and ***rhq.agent.server.bind-port***

The full list of parameters, including advanced setup options, are listed in Table 5.2, "All Options Available During Advanced Setup".

7. Configure the agent as a background service, as in Section 4, "Running the JBoss ON Agent as a Service".

Once the agent is configured, it persists its configuration in the Java Preferences backing store. Once this happens, **agent-configuration.xml** is no longer needed or used. Editing **agent-configuration.xml** will no longer have any effect on the agent, even after restarting the agent. To pick up changes to the **agent-configuration.xml** file, the agent must be restarted with the **--cleanconfig** command line option or the configuration must be reloaded with the **config --import** agent prompt command.

**Important**

If the agent fails to register with the server and seems to hang after outputting the message *The agent does not have plug-ins - it will now wait for them to be downloaded...* or otherwise does not work property after configuring it, please check the agent log file for error messages (*agentInstallDir*/**logs/agent.log**).

Typically, problems occur when the agent binds to an IP address or hostname that is not resolvable or accessible by the JBoss ON servers.

Similarly, make sure all of the JBoss ON servers' public endpoint addresses are resolvable by the JBoss ON agent. The JBoss ON server that is entered for an agent to register with may not be the same one that the agent uses as its primary server; it depends on the high availability configuration. If the agent cannot contact its server, then it fails to start properly.

**Table 5.2. All Options Available During Advanced Setup**

| Setup Option | Description | Normal or Advanced Setup |
|---|---|---|

| Setup Option | Description | Normal or Advanced Setup |
|---|---|---|
| Agent Hostname or IP Address | The address that the binds to to listen for messages from the server. This is usually the same as the address that the JBoss ON server uses to connect to the agent; if the addresses are different because of the network environment, then transport parameters must be set to resolve the address. | Normal |
| Agent Port | The port number that the agent listens on. As with the IP address, this is usually the same as the port configured for the servers to use to connect to agents, but if these ports are different because of the network environment, then transport parameters must be set to resolve the port. | Normal |
| Agent Transport Protocol | Sets the protocol that the agent expects to use to receive incoming messages from the server. This is usually socket or sslsocket. | Advanced |
| Agent Transport Parameters | Sets transport parameters to append to the end of the locator (URL-style address) used by the remoting framework for agent-server connections. | Advanced |
| RHQ Server Hostname or IP Address | Gives the IP address or hostname of the primary server that the agent communicates with. This information *must* be the same as the hostname or IP address that is configured in the JBoss ON server configuration. | Normal |

| Setup Option | Description | Normal or Advanced Setup |
|---|---|---|
| RHQ Server Port | Gives the port number of the primary server that the agent communicates with. This information *must* be the same as the port number that is configured in the JBoss ON server configuration. | Normal |
| RHQ Server Transport Protocol | Sets the transport protocol that the agent uses for outgoing messages to the JBoss ON server. This information *must* be the same as the transport method that the server is configured to expect in its configuration preferences. | Advanced |
| RHQ Server Transport Parameters | Gives additional transport parameters that are to be used when the agent connects to the primary JBoss ON server. Since this is used to connect to the server, these parameters *must* be the same as the transport parameters set in the JBoss ON server configuration. These settings are especially important if the JBoss ON agent needs to connect to a different host or port than what the JBoss ON server actually binds to. | Advanced |

Report a bug

## 3. Silently Installing an Agent

All of the agent configuration is written to and loaded from the **`agent-configuration.xml`** file. During installation, there are some parameters that are predefined and others that are environment-specific and must be provided. The agent setup prompts request this environment or instance-specific information.

However, it is possible to supply that required information before running the agent for the first time. Since all of the required information is supplied, the agent does not prompt for it; it simply starts.

1. Copy the **agent-configuration.xml** file to a working directory. For example:

```
[jsmith@server ~]$ cp agentRoot/rhq-agent/conf/agent-
configuration.xml /tmp/files/
```

2. Uncomment (if necessary) and edit the desired agent parameters.

   Any agent parameters (such as SSL connections and other advanced configuration) can be defined in **agent-configuration.xml**. At a minimum, the entry keys listed in Table 5.3, "Configuration File Keys for Agent Setup" for the agent and server have to be set in the file.

```
[jsmith@server ~]$ vim /tmp/files/agent-configuration.xml

...
<!-- agent properties -->
<entry key="rhq.agent.name" value="agent.example.com"/>
<entry key="rhq.communications.connector.bind-address"
value="255.255.255.1" />
<entry key="rhq.communications.connector.bind-port"
value="16163" />

<!-- server properties -->
<entry key="rhq.agent.server.bind-address"
value="255.255.255.0" />
<entry key="rhq.agent.server.bind-port"        value="7080" />
<entry key="rhq.agent.disable-native-system" value="false"/>
...
```

3. Set the ***rhq.agent.configuration-setup-flag*** key to **true** so that the agent loads it as initial configuration.

```
<entry key="rhq.agent.configuration-setup-flag" value="true" />
```

4. Start the agent, specifying the edited configuration file with the **--config** option>.

```
[jbossadmin@server !]$ agentRoot/rhq-agent/bin/rhq-agent.sh --
config /tmp/files/agent-configuration.xml
```

**Table 5.3. Configuration File Keys for Agent Setup**

| Installer Prompt Text | Key Name | Description |
|---|---|---|
| | rhq.agent.configuration-setup-flag | Tells the installer that the agent configuration is already in the configuration file. This must be set to **true** for the installer to load the configuration file. |

| Installer Prompt Text | Key Name | Description |
|---|---|---|
| [Agent Name] | rhq.agent.name | Gives a unique name to identify the agent to the server. |
| [Agent Hostname or IP Address] | rhq.communications.connector.bind-address | Gives the hostname or IP address that the server will use to connect to the agent. This **<entry>** line may need to be uncommented before it is set. |
| [Agent Port] | rhq.communications.connector.bind-port | Gives the port for the server to use to communicate with the agent. The default (**16163**) can be used in most cases. |
| [RHQ Server Hostname or IP Address] | rhq.agent.server.bind-address | Gives the hostname or IP address that the agent will use to connect to the server to register itself. If this is a hostname, it must be resolvable by the agent. |
| [RHQ Server Port] | rhq.agent.server.bind-port | Gives the port for the agent to use to communicate with the server. The default (**7080**) can be used, assuming the server was configured with the default values. |
| native | rhq.agent.disable-native-system | Enables the JNI libraries used by the agent. This enables the agent to discover and manage some types of resources using the system native libraries. |

Report a bug

## 4. Running the JBoss ON Agent as a Service

In almost all production environments, the agent should be started as a background daemon process. On Windows, this runs as a service. On Linux and Unix systems, the agent starts at boot time from **init.d**.

More detail on editing the agent server wrapper script and managing the agent daemon is covered in the *Administration and Configuration Guide* guide.

Report a bug

## 4.1. Running the Agent as a Windows Service

1. Make sure the agent is fully set up. The agent does not prompt for the configuration when it is started as a service.

2. Edit the **rhq-agent-env.bat** script and set the environment variable to define the system user as whom the init script will run. There are two options:

   - **RHQ_AGENT_RUN_AS** explicitly sets the user account name. This must match the format of a Windows user account name, *DOMAIN\username*.

   - **RHQ_AGENT_RUN_AS_ME** forces the agent to run as whoever the current user is; this uses the format . \ *%USERNAME %*. If both environment variables are defined, this variable overrides **RHQ_AGENT_RUN_AS**.

   > **Note**
   >
   > Before setting **RHQ_AGENT_RUN_AS_ME** or **RHQ_AGENT_RUN_AS**, make sure that the given user actually has permission to start services. If necessary, assign the user the appropriate rights. Assigning rights is covered in the Windows documentation.

   If neither variable is set, the agent init script runs as the local system account (*Default* or .*LocalSystem*).

   Other available environment variables are listed and defined in the comments in the **rhq-agent-wrapper.bat** script.

3. Run the **rhq-agent-wrapper.bat** script to install the init script as a service. Use the **install** command to install the init script.

4. When prompted, fill in the password for the system user as whom the service will run.

Report a bug

## 4.2. Running the Agent as a Daemon or init.d Service

1. Make sure the agent is fully set up. The agent does not prompt for the configuration when it is started as a service.

2. Open the **rhq-agent-env.sh** file.

3. Uncomment and configure the required environment variables for the agent's **bin** directory, the JDK directory, and the PID directory (which must be writable by the agent user).

```
RHQ_AGENT_HOME=agentRoot/rhq-agent/
export RHQ_JAVA_HOME=/usr
RHQ_AGENT_PIDFILE_DIR=/var/run
```

**Note**

When setting the **RHQ_AGENT_PIDFILE_DIR** on Red Hat Enterprise Linux, edit the ***pidfile*** setting in the **rhq-agent-wrapper.sh** script file. The wrapper script value is used by **chkconfig**.

4. Set any of the optional environment variables.
   - **RHQ_AGENT_DEBUG** enables debug logging.
   - **RHQ_AGENT_JAVA_EXE_FILE_PATH** specifies a Java executable.
   - **RHQ_AGENT_JAVA_OPTS** passes settings to the agent JVM.
   - **RHQ_AGENT_ADDITIONAL_JAVA_OPTS** passes additional Java options to the JVM.
5. *Optional.* Configure any custom start commands, as in Section 4.3, “Starting an Agent with a Custom Command”.
6. Log into the system as root.

   **Important**

   The rest of this procedure describes how to configure the agent init script as a service on Red Hat Enterprise Linux. For other Unix systems, follow a similar procedure that corresponds to the specific platform.
7. Make sure the wrapper script is executable.

   ```
   [jsmith@server rhq-agent]$ chmod a+x agentRoot/rhq-agent/bin/rhq-agent-wrapper.sh
   ```
8. Symlink the **rhq-agent-wrapper.sh** file to **/etc/init.d/**. For example:

   ```
   [jsmith@server rhq-agent]$ ln -s agentRoot/rhq-agent/bin/rhq-agent-wrapper.sh /etc/init.d/rhq-agent-wrapper.sh
   ```

> **Important**
>
> On Solaris, symlinking the agent script file requires invoking ***readlink*** in **rhq-agent-wrapper.sh**. ***readlink*** is not supplied by default in some Solaris installations. Solaris users must download ***readlink*** from a source such as Sunfreeware.

9. Register **rhq-agent-wrapper.sh** with **chkconfig**.

```
[root@server rhq-agent] # /sbin/chkconfig --add rhq-agent-wrapper.sh
```

10. Enable the agent service to run at boot time and have it stop gracefully at when the system shuts down.

```
[root@server rhq-agent] # /sbin/chkconfig rhq-agent-wrapper.sh on
```

If the agent service should not be started when the system boots, turn the script off in **chkconfig**:

```
[root@server rhq-agent] # /sbin/chkconfig rhq-agent-wrapper.sh off
```

Report a bug

## 4.3. Starting an Agent with a Custom Command

When configuring the agent to run as a service in Section 4.2, “Running the Agent as a Daemon or init.d Service”, the agent can be configured to start with a custom start command. This is used mainly to start the agent using **su** or **sudo**, allowing the agent to run as a different user.

The start command is defined with the other agent properties in the **rhq-agent-env.sh** file. There are two parts to the configuration: the start command itself and then a setting to enable a password prompt.

```
RHQ_AGENT_START_COMMAND="su -m test -c '${RHQ_AGENT_HOME}/bin/rhq-agent.sh'"
RHQ_AGENT_PASSWORD_PROMPT=true
```

Setting a start command overrides whatever command is passed in the command line to start the agent.

The password prompt may not be required; it depends on the **sudo** and agent user configuration. If it is required, then the password prompt should be enabled so that the user can enter the password or a password should be set in the **RHQ_AGENT_PASSWORD** parameter; otherwise, the start process will appear to hang.

If the defined start command is invalid, then the agent can fail to start. Along with the command formatting, the directory name can be affected; if there are spaces or special characters in the name, those must be escaped for the command to run.

Report a bug

## 5. Changing Agent Connection Configuration

There are two parts to the agent configuration:

- The agent connection properties, which define the agent instance and how it communicates to the server
- The agent JVM properties, which manage agent performance and options

The agent connection properties are defined when the agent is installed. This includes information like the server for it to connect to, its port number, and whether to use SSL connections.

That agent connection configuration is initially read from **`agent-configuration.xml`** and overlaid with the values entered at the setup prompts at start up. After the agent is initially configured, the agent persists that configuration and never refers to the **`agent-configuration.xml`** again. For that information to be changed, the agent connection information has to be wiped out and reset.

To change the connection configuration, one option is to use the **`--cleanconfig`** option and run through the setup wizard again.

```
agentRoot/rhq-agent/bin/rhq-agent.sh --cleanconfig
```

Most JVM and optional settings (the persisted configuration) are made in the **`rhq-agent-env.sh`** file, which is loaded every time the agent starts, or using agent prompt commands like **`setconfig`**.

Report a bug

## 6. About Agent Automatic Updates

Even for automatic upgrades for the agent, certain preparation has to be made to the JBoss ON agent to make sure that the upgrade process goes smoothly and the agent restarts successfully.

Report a bug

### 6.1. The Process When an Agent Autoupgrades

By default, both JBoss ON servers and agents are configured to upgrade agents automatically. As soon as the JBoss ON server is upgraded, then the agents will be upgraded.

> **Note**
>
> The agent should be running in the background to upgrade properly, as in Section 4, "Running the JBoss ON Agent as a Service".

The automatic upgrade process is part of the normal agent tasks of checking the server for updates:

1. The updated server puts the updated agent packages in a directory accessible to the agent.
2. The server notifies the agent that the agent needs to update as soon as the server detects that the agent is running an older version.
3. As the agent prepares to update, it begins shutting down its other process. This can take several minutes, as it gracefully shuts down each thread.
4. The agent downloads the new binaries from the server.
5. The agent spawns a new Java process.
6. The Java process backs up the old agent configuration and applies the update.
7. The Java process then restarts the agent and kills itself.

It is possible to instruct an agent to initiate an update or to check if updates are available using the **`update`** through the agent command line:

```
agentRoot/rhq-agent/bin/rhq-agent.sh

> update
```

Report a bug

## 6.2. Configuring Agent Preferences

Agent preferences for settings like the Java home directory can be set in environment variables for normal use. However, the environment variables set in the shell are lost when the upgrade process stops and restarts the agent, and that means that the agent may not automatically restart after upgrade. Custom settings for the agent, such as **`RHQ_AGENT_HOME`** or **`RHQ_AGENT_ADDITIONAL_JAVA_OPTS`**, should be added to the **`rhq-agent-env.sh`** file. This file is preserved during upgrade so all of the settings are carried over.

**Warning**

Do not edit any of the scripts used to launch the JBoss ON agent. There are four files which should never be modified:

- rhq-agent.sh
- rhq-agent-wrapper.sh
- rhq-agent.bat
- rhq-agent-wrapper.bat

Any changes to the launcher scripts are overwritten during the automatic update. Changes to the environment files (such as **rhq-agent-env.sh** ) are preserved during the update.

Report a bug

## 6.3. Custom Scripts

JBoss ON 3.3 allows for custom scripts to be placed in the agent's **/bin**. When an agent is upgraded, any files with the **.sh** or **.bat** extension found in the agent's **/bin** directory will be copied to the new agent's **/bin** directory.

Report a bug

## 6.4. Upgrading Custom log4j Settings

Any edits to the Java settings in the **rhq-agent-env.sh** file are preserved between upgrades. Also, if any changes are made to the log4j.xml file, the entire file will be copied over.

Report a bug

## 6.5. Upgrading Custom agent-configuration.xml Settings

During the upgrade process, if the current **agent-configuration.xml** differs at all from the newly installed **agent-configuration.xml**, both files are preserved. The current **agent-configuration.xml** file is retained with the same name. The newly installed **agent-configuration.xml** is renamed to **agent-configuration.xml.new**. This allows for custom settings to be retained if the agent is restarted using the **--cleanconfig** option. Users should review both files to ensure that any new settings are added to the **agent-configuration.xml**.

**Note**

If the current **agent-configuration.xml** and the newly installed **agent-configuration.xml** are exactly the same, then an **agent-configuration.xml.new** file will not be created.

Report a bug

## 6.6. Configuring Keystores and Truststores

If the original JBoss ON agent was configured to run over SSL using custom keystores and truststores, then make sure that those stores are configured so that the upgraded agent can still access them:

1. The keystore files must have the word *keystore* in their filenames. For example, **my-agent-keystore.dat**.

2. The truststore files must have the word *truststore* in their filenames. For example, **my-agent-truststore.dat**.

3. Both the keystore and truststore files should be located in the agent's ***agentRoot*/rhq-agent/conf/** directory. Any trust files in the agent's **conf/** directory will be preserved when the agent configuration is cleaned or purged, including during upgrade.

As long as the SSL files are properly set up, then they will be copied over into the new agent configuration, and the new agent will automatically run in SSL.

Report a bug

## 6.7. Setting Write Permissions on the Agent Home Directory

The upgrade process will write new files to the agent's current installation directory, so the agent's system user must have write permissions to the *parent* directory. For example, if the agent is installed in **/opt/rhq/rhq-agent**, then the agent user should have write permissions to the **/opt/rhq** directory.

If necessary, reset the permissions on the agent home directory. For example:

```
[root@server ~]# chown agent_user /opt/rhq
```

Report a bug

## 6.8. Starting the Agent as a Background Service

Section 4, “Running the JBoss ON Agent as a Service” describes how to configure the agent to run as a background service. On Windows, this runs as a service. On Linux and Unix systems, the agent starts at boot time from **`init.d`**.

The auto-upgrade process assumes that the agent is running in the background. If the agent is not running as a background daemon, when the agent auto-updates itself, the old agent process running in the console is shutdown, and the new agent is restarted as a background service if possible.

Report a bug

## 7. Manually Upgrading the JBoss ON Agent

To ensure compatibility with the JBoss ON server, each agent must be upgraded to the same version of JBoss ON as the server.

Agents have the ability to auto-update themselves. So, under most conditions, it isn't necessary to manually upgrade agents. However, if the auto-update fails for some reason or you disabled agent auto-update, then the agent can be upgraded manually.

> **Note**
>
> All agents must be upgraded at the same time. Having agents of different versions is not supported.

1. Shut down the JBoss ON agent.

2. *Windows only*.. If the agent is running as a Windows service, uninstall the Windows service:

   ```
   cd old-agent-install-dir/bin
   ./rhq-agent-wrapper.bat remove
   ```

3. Upgrade the JBoss ON server, as in Chapter 3, *Upgrading JBoss ON Servers and Storage Nodes*. The JBoss ON server must be upgraded before any agents are upgraded.

4. Restart the upgraded JBoss ON servers if they are not yet started.

5. Download the agent update binary from the server.

6. Copy the agent update binary JAR file into the parent directory where the agent is installed. For example:

   ```
   cp agent-update-binary.jar /opt/rhq/rhq-agent
   ```

7. Extract the new JBoss ON agent from the agent update binary by running the following command:

```
java -jar agent-update-binary.jar --
update=agent_installation_directory
```

This will tell the agent update binary to extract the JBoss ON agent distribution and update the current agent that is found in **rhq-agent** subdirectory. At this point, the upgraded JBoss ON agent is located in the original **rhq-agent** directory. The old agent has been backed up to the **rhq-agent-old** directory. Any upgrade errors are written to the agent's log files.

8. Finally, start the JBoss ON agent.

Report a bug

## 8. Reinstalling the Agent

An agent can be re-installed, with completely fresh configuration. There are three points of configuration for the agent: the agent's (local) persisted configuration, the agent inventory (and associated resource data), and the platform entry in the server inventory. Both the configuration on the local machine and the agent and resource configuration on the JBoss ON server need to be cleared for the agent to reinstalled successfully:

- The agent's persisted Java configuration should be purged.
- The agent's inventory should be purged, along with any resource history and configuration.
- The agent must be removed from the JBoss ON inventory. This can be done by deleting the agent from the JBoss ON configuration in the **Administration > Agents** area (preferred) or by removing the platform resource from the inventory.

To reinstall the agent:

1. Make sure that the original agent instance is properly removed.
    a. Stop the agent process.
    b. Remove the platform entry from the JBoss ON server inventory.
2. Restart the agent with the **--fullcleanconfig** option. This registers the agent with a new security token and fresh configuration settings.

```
agentRoot/rhq-agent/bin/rhq-agent.sh --fullcleanconfig
```

Report a bug

## 9. Starting the Agent

The agent is started and runs using a script in the agent's **bin/** directory.

> **Note**
>
> If the agent is installed with a JBoss ON server or a storage node, then it is managed using the **rhqctl** script in the server's **bin/** directory.
>
> All agents installed on managed systems are managed using the **rhq-agent.sh** script.

Unlike the server start script, which starts the server process and then exits the script, the agent script remains open, with a prompt to accept further input if necessary. (Usually, the script can simply be started and left to run in the background.)

```
/opt/rhq-agent/bin/rhq-agent.sh

RHQ 3.3.0-SNAPSHOT [cda7569] (Tue Apr 13 13:39:16 EDT 2015)
>
```

Most of the time, the JBoss ON agent can run without any additional options or settings. Additional configuration options can be set by editing the **rhq-agent-env.sh** script file or by using the **-D** and the property name with the **rhq-agent.sh** script. Changing the settings with the properties file and the start script is covered in the *Configuring JBoss ON Servers, Agents, and Storage Nodes*.

> **Note**
>
> If there are any errors when starting the JBoss ON agent, run the agent start script with the **--cleanconfig** to wipe the previous agent configuration and start fresh.

Report a bug

# Chapter 6. Installing the Agent from RPM

An RPM package is available for Red Hat Enterprise Linux systems to install the agent and, optionally, the agent service.

> **Important**
>
> This chapter is for installing and managing an agent on a platform which will be managed by JBoss ON.
>
> If this system hosts a JBoss ON server, then install the agent as part of the server installation process, as described in Chapter 2, *Installing the JBoss ON Server*. **An agent on the same machine as a JBoss ON server cannot be installed or managed through the agent RPM. It must be installed and managed using the `rhqctl` script.**

Report a bug

## 1. About Agent RPMs

Installing an agent through an RPM offers a simpler and standardized installation process, which makes it possible to install agents on resources in cloud and PaaS environments or when kickstarting machines in data centers.

The JBoss ON agent RPMs can also make it easier to manage the agent more like other Linux system applications because the agent is automatically configured to function as a system service:

- System user and group settings with appropriate permissions already set
- System services to start, stop, and restart the agent
- System service to change the agent's configuration
- Upgrades using system tools

When installing the agent from the JAR file, there are several factors in the install environment that define the agent configuration, such as the installation directory and the location of the Java preferences store. Meaning, the installation directory is determined by where the JAR is unpacked. The agent user and the Java preferences location are both defined by what system user performs the installation.

Many of these settings are defined as part of the RPM setup, so the environment has minimal impact on the resulting agent configuration. This section describes some of the differences between RPM and JAR installations and some of the default configuration set by the RPM process.

Report a bug

## 1.1. Differences Between JAR and RPM Installations

The most notable difference is that the RPM defines the home directory and locations of important files, regardless of the location where the **rpm** is run or the user account (root) who initiated it.

**Table 6.1. Some Differences Between JAR and RPM Installations**

| Configuration Area | JAR Value | RPM Value |
|---|---|---|
| Agent user | Set to the system user who installs it | jbosson-agent user, jbosson group |
| Agent service | Not set | jon-agent |
| Environment variables | *installDir*/bin/rhq-agent-env.sh | /etc/init.d/jon-agent (init script) for ADDITIONAL_JAVA_OPTS<br>/usr/share/jboss-on-3.3.2.GA/agent/bin/rhq-agent-env.sh for JAVA_OPTS |
| Home directory location | Wherever the JAR is installed | /usr/share/jboss-on-3.3.0.GA/agent/ |
| agent-configuration.xml location | In the **conf/** directory where the JAR is installed | /etc/jboss-on/agent/ [a] |
| Java preferences location | ~/.java/default (system user Java preferences) | /var/lib/jboss-on/agent/prefs/.java/.userPrefs/rhq-agent/default/ |
| Data directory location | *agentInstallDir*/data | /var/lib/jboss-on/agent/data/ |
| Log directory location | In the **logs/** directory where the JAR is installed | /var/log/jboss-on/agent/ [b] |
| Autoupgrade | Enabled | Disabled |

[a] symlinked to /usr/share/jboss-on-3.3.0.GA/agent/conf

[b] symlinked to /usr/share/jboss-on-3.3.0.GA/agent/logs

Report a bug

## 1.2. The JBoss ON User

Before installing an agent JAR, one of the most critical decisions is selecting the system user as which the JBoss ON agent will run (Section 1.4, “Picking the Agent System User”). This has security implications for the agent process on the system, and it also affects how the user interacts with local server and application resources — which each have their own system users and permissions.

The agent RPM automatically creates a new system user with the appropriate system configuration to address security issues like directory access.

**Important**

The agent user still has to interact with resources. The appropriate group permissions, SELinux contexts, and other resource configuration can still affect how the agent can discover and manage a resource. Section 1.4, “Picking the Agent System User” outlines these considerations; if necessary, alter the system configuration to allow the agent the appropriate level of access to the resource.

**Note**

The agent RPM creates the **jon-agent** user and the **jbosson** group when it is installed. **The user and group are not removed if the RPM is uninstalled.**

**Table 6.2. Agent User Configuration**

| Property | Value |
|---|---|
| Username | jbosson-agent |
| Group name | jbosson |
| User ID (UID) | 400 |
| Group ID (GID) | 400 |
| User properties | NOSHELL |
| Init script owner | root |
| Init script user | jon-agent [a] |

[a] This can be edited to be any system user.

Report a bug

## 1.3. Service Tools and Init Script

Part of the RPM setup includes configuring the JBoss ON agent as a system service. An init script is installed for the agent at **/etc/init.d/jon-agent**. **chkconfig** is configured so that the agent starts when the system starts and runs as a daemon.

The init script includes all of the normal service management commands, as well as specific commands to manage the agent itself:

- **start**
- **stop**
- **restart**
- **status**
- **kill**, which forces the agent process to stop
- **config**, which runs through the agent configuration wizard again and refreshes the agent configuration with new settings

The **start**, **stop**, **restart**, and **status** commands are available when the agent is manually configured to run as a service, as described in Section 4, "Running the JBoss ON Agent as a Service". However, the **kill** and **config** commands are only available with the init script provided with the agent RPM.

The agent init script, **/etc/init.d/jon-agent**, sets the environment variables that are set in the **rhq-agent-env.sh** file with a JAR installation. This init script defines the agent system user and group, the log and data directory locations, and Java options. Editing the init script can, for example, allow the agent to run as a different user or to start with different JVM settings.

> **Important**
>
> When the agent is installed from the RPM, the only supported way to edit the agent configuration is by running the **config** command or by editing the init script. Editing the **rhq-agent-env.sh** file or other configuration files directly is not supported.

Report a bug

## 1.4. Update Differences

When an agent is installed through a JAR, there is a key set in the **agent-configuration.xml** file that tells the agent to check for upgrades. The agent then polls the server, and if the JBoss ON server version is newer than the agent version, the agent requests updated binaries from the server.

The agent RPMs use an entirely different installation path than the agent JAR files, and an agent installed as an RPM relies on the local system tools to manage its packages. The upgrade flag, then, in the **agent-configuration.xml** file is turned off, to disable attempts at an autoupgrade and to allow the local system to manage the agent packages.

```
<entry key="rhq.agent.agent-update.enabled" value="false" />
```

With autoupdates disabled, the agent must be upgraded manually whenever the JBoss ON server is upgraded, to ensure that its version remains in sync with the JBoss ON server version.

Report a bug

### 1.5. Available Channels

**Table 6.3. Available Channels for the Agent RPM**

| Product Name | Product Version | Channel Name |
|---|---|---|
| JBoss Enterprise Application Server (EAP) | 5, x86 | jbappplatform-5-i386-server-6-rpm |
| JBoss Enterprise Application Server (EAP) | 5, x86_64 | jbappplatform-5-x86_64-server-6-rpm |
| JBoss Enterprise Application Server (EAP) | 6, x86 | jbappplatform-6-i386-server-6-rpm |
| JBoss Enterprise Application Server (EAP) | 6, x86_64 | jbappplatform-6-x86_64-server-6-rpm |

Report a bug

## 2. Installing the Agent from RPM

The JBoss ON agent is installed through the **jboss-on-agent** package. This package installs all of the agent files, creates a specific JBoss ON agent system user, and configures the JBoss ON agent as a system service.

> **Important**
>
> This procedure is for installing and managing an agent on a platform which will be managed by JBoss ON.
>
> If this system hosts a JBoss ON server, then install the agent as part of the server installation process, as described in Chapter 2, *Installing the JBoss ON Server*. **An agent on the same machine as a JBoss ON server cannot be installed or managed through the agent RPM. It must be installed and managed using the rhqctl script.**

Report a bug

## 2.1. Installing Using yum

1. Configure the **yum** repos to include the JBoss ON channel (as listed in Section 1.5, "Available Channels"). For example:

```
[root@server ~]# yum-config-manager --enable jbappplatform-6-
x86_64-server-6-rpm
```

> **Note**
>
> Installing the RPM requires specific entitlements for the RHN user account for the 3.3 release. This RHN user account must be used to register the system to have access to the appropriate repositories.

2. Use **yum** to install the package.

```
[root@server ~]# yum install jboss-on-agent
```

This installs the agent in **/usr/share/jboss-on-3.3.2.GA/agent**.

3. Configure the agent by running the **service jon-agent config** command. This runs through the advanced installer to configure the agent.

```
[jsmith@server ~]$ sudo service jon-agent config
RHQ 4.4.0.JON311GA [6910991] (Wed Aug 01 18:43:03 EDT 2012)
** Advanced Setup **
... 8< ...
Agent Name [agent.example.com] : agent1
Agent Hostname or IP Address [!*] :
Agent Port [16163] :
Agent Transport Protocol [socket] :
Agent Transport Parameters
[numAcceptThreads=1&maxPoolSize=303&clientMaxPoolSize=304&soc
ketTimeout=60000&enableTcpNoDelay=true&backlog=200] :
RHQ Server Hostname or IP Address [255.255.255.255] :
RHQ Server Port [7080] :
RHQ Server Transport Protocol [servlet] :
RHQ Server Transport Parameters [/jboss-remoting-servlet-
invoker/ServerInvokerServlet] :
RHQ Server Alias [rhqserver] :
The setup has been completed for the preferences at node
[/rhq-agent/default].
```

The **config** command runs through all of the advanced setup options, which configures three areas of the agent:

- *The agent connection information*, used to register the agent to the server

- The agent name
- The agent port
- The agent host (by hostname or IP address)

- *The agent-server communication settings*, which include configuring SSL or secure connections and rules on how frequently the agent communicates with the agent
  - The agent protocol, either socket (regular) or sslsocket (secure)

    For sslsocket, the JBoss ON server needs to be configured to accept SSL connections, as in the SSL chapter of *Configuring JBoss ON Servers and Agents*.

  - Any client transport parameters to use to connect to the server

    Both the server and the agent use JBoss Remoting for communication. JBoss Remoting allows servers and clients to pass connection settings using a URL-style address. Transport parameters include things like pool sizes, timeout periods, and buffer settings. For the complete list, see the JBoss Remoting client parameters documentation.

  - The server protocol which the agent uses to send messages to the server, either servlet (regular) or sslservlet (secure)

    The server connection settings configured on the agent *must* match the configuration in the server itself.

  - Any server transport parameters to use to receive messages from the agent

    Both the server and the agent use JBoss Remoting for communication. JBoss Remoting allows servers and clients to pass connection settings using a URL-style address. Transport parameters for the server relate to the servlet used to receive agent messages.

- *The JBoss ON server to register with*
  - The server host (by hostname or IP address)
  - The server port
  - The server alias, a short nickname to identify the server instance

4. Start the agent.

```
[jsmith@server ~]$ sudo service jon-agent start
```

Report a bug

## 2.2. Installing by Downloading the RPM

1. Log into the Customer Portal at https://access.redhat.com.

2. Click the **Downloads** tab.

3. Click the **Packages** box to search for the agent RPM package.

4. Enter **jboss-on-agent** as the package name, select the **In the following architectures** radio button, and set the appropriate architecture for the system.

5. Click the package name, and then click the name for the latest agent RPM update.

6. Scroll to the bottom of the page, and click the **Download Package** link and save the package to a convenient location on the system.

7. Install using the **rpm** command. For example:

```
[root@server ~]# rpm -ivh /tmp/downloads/jboss-on-agent-
3.3.2.GA.el6.noarch.rpm
```

   This installs the agent in **/usr/share/jboss-on-3.3.2.GA/agent**.

8. Configure the agent by running the **service jon-agent config** command. This runs through the advanced installer to configure the agent.

```
[jsmith@server ~]$ sudo service jon-agent config
RHQ 4.4.0.JON311GA [6910991] (Wed Aug 01 18:43:03 EDT 2012)
** Advanced Setup **
... 8< ...
Agent Name [agent.example.com] : agent1
Agent Hostname or IP Address [!*] :
Agent Port [16163] :
Agent Transport Protocol [socket] :
Agent Transport Parameters
[numAcceptThreads=1&maxPoolSize=303&clientMaxPoolSize=304&soc
ketTimeout=60000&enableTcpNoDelay=true&backlog=200] :
RHQ Server Hostname or IP Address [255.255.255.255] :
RHQ Server Port [7080] :
RHQ Server Transport Protocol [servlet] :
RHQ Server Transport Parameters [/jboss-remoting-servlet-
invoker/ServerInvokerServlet] :
RHQ Server Alias [rhqserver] :
The setup has been completed for the preferences at node
[/rhq-agent/default].
```

The **`config`** command runs through all of the advanced setup options, which configures three areas of the agent:

- *The agent connection information*, used to register the agent to the server
  - The agent name
  - The agent port
  - The agent host (by hostname or IP address)
- *The agent-server communication settings*, which include configuring SSL or secure connections and rules on how frequently the agent communicates with the agent
  - The agent protocol, either socket (regular) or sslsocket (secure)

    For sslsocket, the JBoss ON server needs to be configured to accept SSL connections, as in the SSL chapter of *Configuring JBoss ON Servers and Agents*.
  - Any client transport parameters to use to connect to the server

    Both the server and the agent use JBoss Remoting for communication. JBoss Remoting allows servers and clients to pass connection settings using a URL-style address. Transport parameters include things like pool sizes, timeout periods, and buffer settings. For the complete list, see the JBoss Remoting client parameters documentation.
  - The server protocol which the agent uses to send messages to the server, either servlet (regular) or sslservlet (secure)

    The server connection settings configured on the agent *must* match the configuration in the server itself.
  - Any server transport parameters to use to receive messages from the agent

    Both the server and the agent use JBoss Remoting for communication. JBoss Remoting allows servers and clients to pass connection settings using a URL-style address. Transport parameters for the server relate to the servlet used to receive agent messages.
- *The JBoss ON server to register with*
  - The server host (by hostname or IP address)
  - The server port
  - The server alias, a short nickname to identify the server instance

9. Start the agent.

```
[jsmith@server ~]$ sudo service jon-agent start
```

Report a bug

## 3. Changing the Agent Configuration After an RPM Install

There are two parts to the agent configuration:

- The agent connection properties, which define the agent instance and how it communicates to the server
- The agent JVM properties, which manage agent performance and options

Report a bug

### 3.1. Changing Agent Connection Configuration

The agent connection properties are defined when the agent is installed. This includes information like the server for it to connect to, its port number, and whether to use SSL connections.

That agent connection configuration is initially read from **`agent-configuration.xml`** and overlaid with the values entered at the setup prompts at start up. After the agent is initially configured, the agent persists that configuration in its Java preferences (**`/var/lib/jboss-on/agent/prefs/.java/.userPrefs/rhq-agent/default/`**) and never refers to the **`agent-configuration.xml`** again.

For that information to be changed, the agent connection information has to be wiped out and reset.

To change the agent connection (registration) configuration, use the **`config`** service command to run through the setup options again. This cleans out the preferences store, re-reads the **`agent-configuration.xml`** file, and runs through the configuration setup again.

> **Important**
>
> When the JBoss ON agent is installed from an RPM, it is automatically configured as a system service. For data consistency and agent performance, always manage the agent connection configuration using the service tools, rather than attempting to edit the configuration files or JVM startup properties directly.

For example:

```
[jsmith@server ~]$ sudo service jon-agent config
```

```
RHQ 4.4.0.JON311GA [6910991] (Wed Aug 01 18:43:03 EDT 2012)
** Advanced Setup **
Agent Name [agent.example.com] : agent1
Agent Hostname or IP Address [!*] :
Agent Port [16163] :
Agent Transport Protocol [socket] :
... 8< ...
```

The **config** service command opens the agent start script and automatically passes a series of options which edit the agent connection configuration:

- **--cleanconfig**, to wipe the previous configuration settings
- **--setup** and **--advanced**, which force the agent to run its configuration setup again
- **--daemon** and **--nostart**, which runs the agent command prompt without starting the agent process and then exits (so that the agent can be started as a service)

So, the **service jon-agent config** command is equivalent to starting the agent with all those options:

```
rhq-agent.sh --cleanconfig --setup --advanced --daemon --nostart
```

Report a bug

### 3.2. Changing Agent JVM and Other Init Configuration

After the agent is configured, optional JVM settings (the persisted configuration) are set in the init script, **/etc/init.d/jon-agent** file, or in the environment script, **rhq-agent-env.sh**. Both files are loaded every time the agent starts; it is recommended to edit the init script, which sets the additional JAVA_OPTS values.

For example:

```
RHQ_AGENT_ADDITIONAL_JAVA_OPTS="-Drhq.agent.data-
directory=$RHQ_AGENT_DATA_DIR -
Djava.util.prefs.userRoot=$RHQ_AGENT_PREFS_DIR -Xms64m -Xmx128m -
Djava.net.preferIPv4Stack=true"
export RHQ_AGENT_ADDITIONAL_JAVA_OPTS
```

The Java settings can also be edited using agent prompt commands like **setconfig**. This is described in Configuring JBoss ON Servers and Agents.

Report a bug

## 4. Migrating from a JAR Installation to an RPM Installation

When an agent is initially installed from the JAR file available with the JBoss ON server downloads (Chapter 5, *Installing and Upgrading an Agent on a Managed Platform from the JAR File*), it is possible to switch the agent to an RPM installation. There are two potential paths. Either the old agent can be scrapped and a new agent installed (which loses all of the original agent configuration and data) or the agent data can be migrated from the original JAR installation to the new RPM installation.

Report a bug

## 4.1. Converting an Agent (Losing Configuration Data)

This method loses all of the original data for the agent, including the persistent store, data directory, and logs. However, by re-using the original security token, the agent can re-register with the server and retrieve its previous inventory and resource histories and configuration, so none of the management information for the platform is lost.

1. Retrieve the security token for the agent.

    a. Click the **`Administration`** tab and select the **`Agents`** link under the **`Topology`** section on the left.

    b. Select the agent from the list, and click its name to open its details page.

Agents
« Back to List
Details
Name : localhost1
Agent Bind Address : 10.0.0.1
Remote Endpoint : socket://10.0.0.1:16163/?
rhq.communications.connector.rhqtype=agent&numAcceptThreads=1&maxPoolSize=303&clientMaxPoolSize=304&socketTim
Agent Bind Port : 16163
Token : ISxZBHecrMlYPkNEAPpkRyDQ9egr7bXwQ3wTrvuJ3hvQArZoVyHpwEh+LyAdDWksfLo=
Last Availability Report : October 1, 2014 12:28:17 PM UTC-5
Last Availability Ping : October 1, 2014 2:43:18 PM UTC-5
Affinity Group : South Data Center
Connected Server : localhost1

    c. Copy the security token.

2. Shut down the agent.

3. Remove the JAR installation directory.

4. Install the agent RPM.

5. Edit the **`agent-configuration.xml`** file and add a line for the original security token for the agent.

```
vim /etc/jboss-on/agent/agent-configuration.xml

<entry key="rhq.agent.security-token" value="abcd1234" />
```

6. Run through the agent configuration installer.

```
[jsmith@server ~]$ sudo service jon-agent config
```

7. Start the agent.

```
[jsmith@server ~]$ sudo service jon-agent start
```

Report a bug

## 4.2. Migrating an Agent to an RPM (Preserving Configuration Data)

It is possible to preserve the JVM and persisted data for the agent, which maintains all of the original configuration data. However, this requires accessing the Java store through a Java preferences browser in the original agent and copying it into the Java store for the new agent, without affecting any other data in the store. There is always a risk when editing Java stores.

1. Shut down the original agent.

2. Install the agent RPM.

3. Copy over the previous configuration directories for the agent. This includes the data directory (which contains operational information like the changesets for drift detection or truststores used for SSL) and the log directory. For example:

```
[root@server ~]# cp -r agentRoot/rhq-agent/data/
/var/lib/jboss-on/agent/data/
[root@server ~]# cp -r agentRoot/rhq-agent/logs/
/var/log/jboss-on/agent/
```

4. Using a Java preferences editor, export the Java preferences specific to the agent from the original preferences store in **`~/.java/.userPrefs/rhq-agent/default`** (by default).

> **Note**
>
> Be sure to retrieve the security token. The token allows the agent to re-register with the server successfully.

5. Using a Java preferences editor, import the Java preferences for the agent into the new preferences store in **`/var/lib/jboss-on/agent/prefs/default`** (by default).

6. Run through the agent configuration installer.

```
[jsmith@server ~]$ sudo service jon-agent config
```

7. Start the agent.

```
[jsmith@server ~]$ sudo service jon-agent start
```

Report a bug

## 5. Starting the Agent

By default, when the agent is installed using an RPM, it is configured as a system service. This means that it is started using the **service** command:

```
[root@@server ~]# service jon-agent start
```

The agent can also be started and runs using a script in the agent's **bin/** directory.

> **Note**
>
> If the agent is installed with a JBoss ON server or a storage node, then it is managed using the **rhqctl** script in the server's **bin/** directory.
>
> **Any agent installed through an RPM cannot be managed with the rhqctl script.**

The agent command prompt is useful for passing certain agent commands and changing agent configuration. Using the **rhq-agent.sh** command opens the command prompt.

```
/opt/rhq-agent/bin/rhq-agent.sh

RHQ 3.3.0-SNAPSHOT [cda7569] (Tue Apr 13 13:39:16 EDT 2015)
>
```

Report a bug

## 6. Upgrading the Agent RPM

The RPM with the agent packages installs an agent on a managed platform. That agent can then be upgraded using the latest agent RPM.

If an agent was installed on the same system as a JBoss ON server, that agent *cannot* be upgraded or managed through the agent RPM or the configured system tools. That agent must be converted to work with the JBoss ON server and its **rhqctl** tool.

Report a bug

## 6.1. Upgrading the Agent RPM on a Managed System

> **Important**
>
> This section is for upgrading an agent on a platform which will be managed by JBoss ON.
>
> If this system hosts a JBoss ON server, then migrate the agent as part of the server installation process, as described in Section 6.2, "Migrating an Agent on a JBoss ON Server Machine". **An agent on the same machine as a JBoss ON server cannot be installed or managed through the agent RPM. It must be installed and managed using the `rhqctl` script.**

1. Configure the **`yum`** repos to include the JBoss ON channel. For example:

```
[root@server ~]# yum-config-manager --enable jbappplatform-6-x86_64-server-6-rpm
```

2. Use **`yum`** to upgrade the package.

```
[root@server ~]# yum upgrade jboss-on-agent
```

It is also possible to download the packages from the Customer Portal and then upgrade them using the **`rpm -U`** command:

```
[root@server ~]# rpm -Uvh jboss-on-agent-3.3.2.GA.el6.noarch.rpm
```

Report a bug

## 6.2. Migrating an Agent on a JBoss ON Server Machine

In JBoss ON 3.1.x versions, an agent was installed and managed independently of any installed JBoss ON server, even if they were on the same system. This meant that an agent could be installed from an RPM on a system which hosted a JBoss ON server, as well as a managed platform.

However, starting in JBoss ON 3.2, any agent installed on the same system as a JBoss ON server is installed and managed in tandem with the server, using the same installation packages and management script (**`rhqctl`**). **This means that agents on a JBoss ON server machine cannot be installed from an RPM.**

For existing agent installations, the agent must be migrated to the new management scripts and configuration.

**Note**

This migration script is provided as a convenience for skilled administrators. For assistance, contact Red Hat support services.

1. Install a new JBoss ON 3.3 server or upgrade a server to JBoss ON 3.3.

2. Get the agent name, token, bind address, and server bind address for the agent.

    a. Log into the JBoss ON server UI.

    b. Click the **Administration** tab in the top menu.

    c. In the **Topology** box on the left, click the **Agents** link.

    d. Click the name of the agent to be migrated in the list.

    e. The **Agent Details** area lists all of the required information. This agent information must be coped into the **agent-configuration.xml** file to migrate the agent instance.

Agents
« Back to List
Details
Name : localhost1
Agent Bind Address : 10.0.0.1
Remote Endpoint : socket://10.0.0.1:16163/?
rhq.communications.connector.rhqtype=agent&numAcceptThreads=1&maxPoolSize=303&clientMaxPoolSize=304&socketTim
Agent Bind Port : 16163
Token : ISxZBHecrMIYPkNEAPpkRyDQ9egr7bXwQ3wTrvuJ3hvQArZoVyHpwEh+LyAdDWksfLo=
Last Availability Report : October 1, 2014 12:28:17 PM UTC-5
Last Availability Ping : October 1, 2014 2:43:18 PM UTC-5
Affinity Group : South Data Center
Connected Server : localhost1

3. Copy the existing **agent-configuration.xml** file to the new agent location, ***agentRoot*/rhq-agent/conf/agent-configuration.xml**.

4. Update the **agent-configuration.xml** properties for the agent identity information. For example:

```
<entry key="rhq.agent.name" value="agent-01" />
<entry key="rhq.agent.security-token" value="abcd1234" />
<entry key="rhq.agent.server.bind-address"
value="server.example.com" />
<entry key="rhq.communications.connector.bind-address"
value="1.1.1.1" />
```

| UI Field Name | Configuration File Property |
|---|---|
| Agent Name | rhq.agent.name |
| Token | rhq.agent.security-token |
| Address | rhq.agent.server.bind-address |
| Current Server | rhq.communications.connector.bind-address |

5. Stop the old agent process.

```
[jsmith@server ~]$ sudo service jon-agent stop
```

6. Copy this example script, and fill in the location for the old agent installation, the updated configuration file, and the new server in the **`agent-configuration-migrate.sh`** script.

```
###### agent-configuration-migrate.sh #########
#!/bin/sh
#
#
# Note: Assumes and installs agent into default location. Modify
steps as
necessary
#       if this is not true.
##################################################################
#######

#Ex. OLDER_RPM_AGENT_INSTALL=/usr/share/jboss-on-3.3.0.GA/agent
OLDER_RPM_AGENT_INSTALL=

#Ex. AGENT_MIGRATION_CONFIG_LOCATION=/tmp/agent-migration.xml
For security
consider using $(mktemp) if you automate this further.
AGENT_MIGRATION_CONFIG_LOCATION=

#Ex. NEWEST_SERVER_LOCATION=/opt/jon/jboss-on-3.3.1.GA . Note
bin, etc,
modules are immediate sub directories.
NEWEST_SERVER_LOCATION=

#Install newer native agent including older agent configuration
details.
#NOTE: new agent will be installed to default location. Modify
the following
line accordingly
$NEWEST_SERVER_LOCATION/bin/rhqctl install --agent --agent-
config
$AGENT_MIGRATION_CONFIG_LOCATION

 # Echo next steps to complete migration.
 echo -e "\n If no errors, then migration of older agent
configuration was
```

```
 successful."
 echo Ex. Additional environment variables added to old agent.
 echo -e "\t i) (if necessary) Merge
$OLDER_RPM_AGENT_INSTALL/bin/*.sh with
 $NEWEST_SERVER_LOCATION/../rhq-agent/bin/*.sh."
 echo -e "\t ii)(if necessary) Manually and carefully merge old
and new agent
 log settings."
 echo -e "\t iii)Continue JON server upgrade. Ex. \n '
 $NEWEST_SERVER_LOCATION/bin/rhqctl upgrade --from-server-dir
(insert older
 jon server directory) --run-data-migrator'"
 echo    "iv)Start all desired components. Ex.'
 $NEWEST_SERVER_LOCATION/bin/rhqctl start'"
 echo    "v)Verify migration and remove intermediate migration
 scripts/files.'"
 echo "Done."
```

7. Run the migration script.

Report a bug

### 6.3. Upgrading Custom agent-configuration.xml and log4j.xml Settings with RPMs

If **`agent-configuration.xml`** or **`log4j.xml`** are edited by a user, then the RPM upgrade process retains those files and adds the newest versions of these files as *.rpmnew. For more detail about the RPM upgrade process see the RPM section of the Red Hat Enterprise Linux System Administration Guide.

Report a bug

## 7. Troubleshooting RPM Installs

**Q:** **I installed my agent successfully, but now it won't connect to the server. Why?**

**A:** The agent has to be started in the foreground to run through its installer. The installer is where the agent is told what JBoss ON server to access. If the agent service is started *without* going through the agent installer, the agent loads its configuration without ever identifying the server to connect to.

The agent configuration has to be edited, as described in Section 3, “Changing the Agent Configuration After an RPM Install”, to set the server connection information.

```
[jsmith@server ~]$ sudo service jon-agent config
```

Report a bug

# Chapter 7. Installing an Agent Using the JBoss ON User Interface

The JBoss ON UI provides the ability to install a JBoss ON agent on a server over SSH. This can be used on server with existing JBoss ON agents as well as servers without any JBoss ON agents.

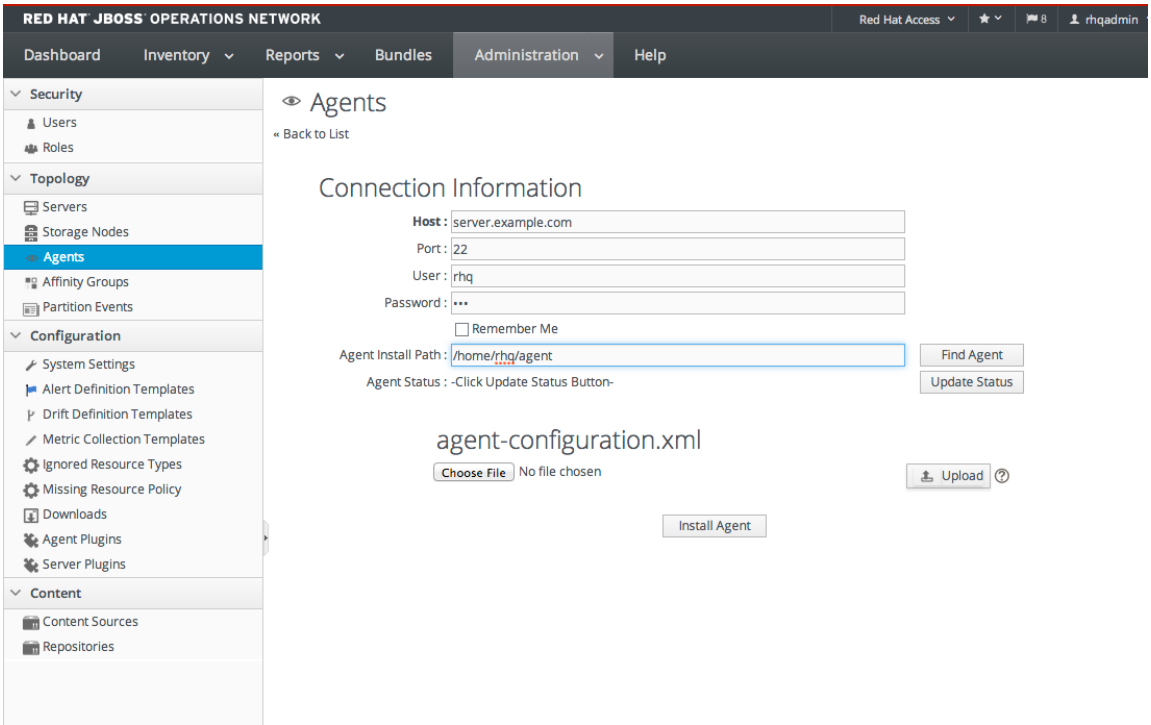
> **Note**
>
> The credentials used to perform the remote install are the same used to start the JBoss ON agent.

To install the agent using the JBoss ON UI:

1. Click on the **`Administration`** menu in the top navigation bar, then click on **`Agents`** under the **`Topology`** section in the left navigation.

2. Click on the **`New`** button at the bottom.

3. The fill in the **`host, port, user,`** and **`password`** information. For more details on each field, please see Section 1, “Installation Options”

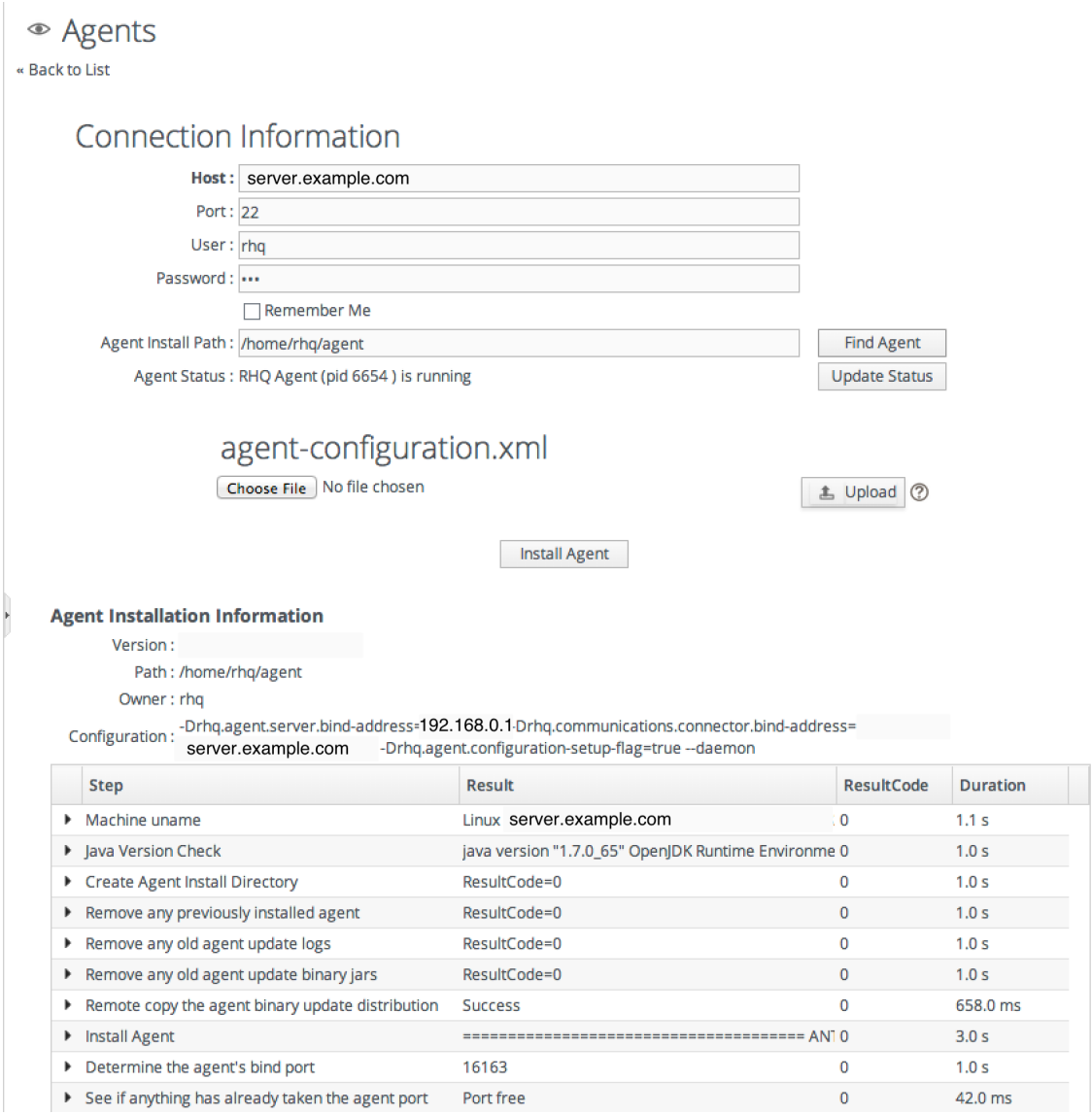



4. Fill in the desired installation path for the agent. The **`Find Agent`** button may also be used search below the path specified in **`Agent Install Path`**. For more details on **`Agent Install Path`**, please see Section 1, “Installation Options”.

5. Optional. An **`agent-configuration.xml`** may be specified for the newly installed agent to use. If no **`agent-configuration.xml`** is set, it will use the default.

6. Click **`Install Agent`**.

7. Once the agent install is complete, the agent installation information is shown. Additionally, **`Agent Status`** will be updated to reflect the state of the newly installed agent and the agent can also be started and stopped using the corresponding buttons.




Report a bug

## 1. Installation Options

The JBoss ON Agent installation over SSH provides the following fields for the installation process:

- **`Host`** - This field specifies the hostname of the server to connect to using SSH.

- **`Port`** - This field specifies the port to use for the SSH connection. By default, it will use port 22.

- **User and Password** - These fields are the credentials used to connect to the server. This user will also be used to start and stop the JBoss ON agent.

- **Agent Install Path** - This field specifies the path where the JBoss ON agent will be installed. The JBoss ON UI also provides the **Find Agent** button, which will attempt to search for any existing agents that are installed below the parent path specified by the **Agent Install Path** field. If an agent is found under the specified path, **Agent Install Path** will be updated with the path to that agent. The found agent can then also be started and stopped using the corresponding buttons.

  If nothing is entered, common locations are searched on the host for an installation path.

  > **Note**
  >
  > To perform the installation, the user specified in the **User** field will need permissions on the path specified in **Agent Install Path**. If **Agent Install Path** corresponds to an existing agent and an installation is attempted, the installation process will warn the user prior to overwriting the existing agent.

- **Update Status** - This field provides a status of the existing agent in **Agent Install Path** or the installation process once the **Install Agent** button is clicked.

- **Agent-Configuration.xml** - This field allows users to specify a custom **agent-configuration.xml** file to be used. This field is optional. If no **agent-configuration.xml** file is specified, then the default will be used.

Report a bug

# Chapter 8. Installing and Removing JBoss Agent Plug-in Packs

JBoss Operations Network has additional agent plug-ins to handle specific JBoss resources — EWS, EDS, EAP, or SOA-P. Although these are JBoss ON resource plug-ins, they are included in separate packages and require a separate subscription to download them.

When plug-ins are installed on the server, they are downloaded by and available to all configured JBoss ON agents. This means that the JBoss resources can be discovered and imported into the inventory to be managed,

Plug-ins can be removed on all JBoss ON agents by removing the plug-in from the plug-in directory and restarting the server, or through resource management through the JBoss ON Management Console.

Report a bug

## 1. Installing JBoss Agent Plug-in Packs

**Procedure 8.1. How To Install JBoss ON Agent Plug-in Packs using the Command Line or GUI**

1. Download the plug-in JAR files.

    a. Load the JBoss ON Software Download page and log on using your Red Hat Network credentials.

    b. Open the **Product** drop-down, and select the **JBoss ON for** *Plug-in* product in the drop-down box.

    c. Select the **JBoss ON for** *Plug-in* product in the drop-down box.

    d. Download the required plug-in packs.

2. Extract the plug-in pack archives to a temporary location. This creates a subdirectory with the name **jon-plugin-pack-***plugin_name-version*. For example:

```
[jsmith@server rhq-agent]$ unzip jon-plugin-pack-eap-3.3.zip -d /tmp
```

3. Copy the extracted plug-in JAR files from the **jon-plugin-pack-*plugin_name*-3.3/** directory to the JBoss ON server plug-in directory. For example:

```
[root@server rhq-agent]# cp /tmp/jon-plugin-pack-plugin_name-3.3/*.jar /opt/jon/jon-server-3.3.2.GA/plugins
```

4. **Standalone Platforms**

Have the JBoss ON server update its plug-ins. This can be done through the JBoss ON GUI or by restarting the server.

To load the plug-ins using the GUI:

a. Open the **Administration** tab.

b. In the **Configuration** area on the left, select the **Agent Plug-ins** link.

c. At the bottom of the list of loaded agent plug-ins, click the **SCAN FOR UPDATES** button.

5. **Managed Platforms**

All agents installed on managed platforms must update their plug-ins to use the newly-installed JBoss plug-ins. The agents can manually reload their plug-ins to load the new plug-ins from the agent's command prompt using the **plugins** command:

```
[jsmith@server ~]$ agentRoot/rhq-agent/bin/rhq-agent.sh
> plugins update
```

Alternatively, if the agents have been imported into the JBoss ON inventory, this can be done in the JBoss ON GUI by scheduling an *update plugins* operation for an agent or a group or agents. Select the agent resource entry in the inventory, open the **Operations** tab, and schedule the update plug-ins operation.

Report a bug

## 2. Removing JBoss Agent Plug-in Packs

**Procedure 8.2. How to Remove a JBoss ON Plug-in Pack Manually**

1. Open /opt/jon/jon-server-3.3.2.GA/plugins

2. Delete the plug-in that you no longer want in inventory.

3. **Standalone Platforms**

Have the JBoss ON server update its plug-ins. This can be done through the JBoss ON GUI, or by restarting the server.

To reload the plug-ins through the GUI:

a. Open the **Administration** tab.

b. In the **Configuration** area on the left, select the **Agent Plug-ins** link.

c. At the bottom of the list of loaded agent plug-ins, click the **SCAN FOR UPDATES** button.

4. **Managed Platforms**

All agents installed on managed platforms must update their plug-ins to recognize the deleted JBoss plug-in. The agents can manually reload their plug-ins to detect changes from the agent's command prompt using the **plugins** command:

```
[jsmith@server ~]$ agentRoot/rhq-agent/bin/rhq-agent.sh
> plugins update
```

Alternatively, if the agents are imported into the JBoss ON inventory, this can be done in the JBoss ON GUI by scheduling an *update plugins* operation for an agent or a group or agents.

To schedule an Update Plug-ins Operation:

a. Select the agent resource entry in the inventory

b. Open the **Operations** tab

c. schedule the update plug-ins operation

Report a bug

# Chapter 9. Installing the JBoss ON CLI

The JBoss Operations Network CLI is a standalone Java application that uses the Java Scripting API (this requires Java 6 or later). The CLI allows JBoss ON to be better integrated into the network environment by allowing administrators to interact with JBoss ON programmatically.

A large subset of JBoss ON functionality is exposed in the CLI. This *remote client* (the CLI) is downloaded from the JBoss ON server as an archive, **`rhq-remoting-cli-4.12.0.JON330GA.zip`**.

> **Note**
>
> Only the corresponding version of the CLI can be used to manage the JBoss ON server. Other versions are not compatible.

To install the CLI:

1. Open the JBoss ON GUI.

   ```
   http://server.example.com:7080
   ```

2. Click the **`Administration`** link in the main menu.

3. Select the **`Downloads`** menu item.

4. Scroll to the **`Command Line Client Download`** section, and click **`Download Client Installer`**.

5. Save the **`.zip`** file into the directory where the CLI should be installed.

6. Unzip the packages.

   ```
   [root@server jon]# unzip rhq-remoting-cli-4.12.0.JON330GA.zip
   ```

Report a bug

# Chapter 10. Troubleshooting Installation and Upgrade

This chapter looks at some of the most common issues that may be encountered after installing or upgrading the JBoss Operations Network server. Other issues are also covered in the JBoss ON frequently-asked-questions.

## 10.1. Exceptions and Error Logs

**Q: I'm seeing null pointer exceptions for the `org.apache.catalina.connector.CoyoteAdapter` service. What do these mean?**

**A:** Null pointer exceptions for the **`org.apache.catalina.connector.CoyoteAdapter`** service are returned when the JBoss ON 3.3 server is first installed. These errors are harmless and can be ignored. Installation will complete successfully, and both the server and the GUI will start and run properly.

**Q: I upgraded to 3.3, and there are null pointer exceptions (javax.management.InstanceNotFoundException) in my error logs about the transport service not being registered.**

**A:** When starting a server while agents are running, the server may log transport servlet errors in the logs:

```
[org.rhq.enterprise.server.resource.metadata.ResourceMetadataManag
erBean]
Persisting new ResourceType [ModeShapePlugin:Sequencing
Service(id=0)]...
2011-01-10 16:45:38,571 ERROR
[org.apache.catalina.core.ContainerBase]
Servlet.service() for servlet ServerInvokerServlet threw exception
java.lang.reflect.UndeclaredThrowableException
 at $Proxy424.processRequest(Unknown Source)
 at
org.jboss.remoting.transport.servlet.web.ServerInvokerServlet.proc
essRequest(ServerInvokerServlet.java:128)
 at
org.jboss.remoting.transport.servlet.web.ServerInvokerServlet.doPo
st(ServerInvokerServlet.java:157)
 at javax.servlet.http.HttpServlet.service(HttpServlet.java:710)
 at javax.servlet.http.HttpServlet.service(HttpServlet.java:803)
....
```

This is because the remoting (communications or transport) classes are loaded early in the startup sequence, before the server is completely started. This causes some communications interruptions until the server is completely started. These errors can be ignored.

**Q: I'm seeing error messages when I install (or upgrade) my server. What do they mean?**

**A:** During the upgrade, you may see error messages in the console similar to the following:

```
ERROR [ClientCommandSenderTask] {ClientCommandSenderTask.send-
failed}Failed to send
command [Command: type=[remotepojo]; cmd-in-response=[false];
config=[{rhq.timeout=1000,
rhq.send-throttle=true}]; params=
[{targetInterfaceName=org.rhq.enterprise.communications.Ping,
invocation=NameBasedInvocation[ping]}]]. Cause:
org.jboss.remoting.CannotConnectException:[.....]
```

These can be ignored.

**Q: I upgraded to JBoss ON 3.0.1. However, I see null pointer exceptions in my server logs and the plug-ins still show version 3.0.0. (The 'Server Name' field was changed during upgrade.)**

**A:** During upgrade, the process to register the upgraded server plug-ins fails with a null pointer exception. For example:

```
2012-03-08 20:33:34,523 ERROR
[org.rhq.enterprise.server.core.plugin.ServerPluginScanner]
Failed to register server plugin file [/home/hudson/jon-server-
3.0.1.GA/jbossas/server/default/deploy/rhq.ear/rhq-
serverplugins/rhq-serverplugin-ant-bundle-
4.2.0.JON.3.0.1.GA.jar]
java.lang.NullPointerException
 at
org.rhq.enterprise.server.core.plugin.ServerPluginScanner.regis
terServerPlugin(ServerPluginScanner.java:212)
...
```

This error only occurs if a different server name was entered in the configuration page when the server was upgraded. Changing the **Server Name** field **is not supported** for upgrades.

**Q: The error log is showing ErrorCode=[2289]. Why?**

**A:** With Oracle, selecting the overwrite tables option when there is nothing to overwrite causes an error message with **ErrorCode=[2289]**. This can be ignored.

## 10.2. Connection Issues

**Q: I can't connect to my server after installation.**

**A:** If the installer is not bound to 0.0.0.0 when setting up a server, then it does not set all of the required connection properties. Specifically, the installer does not set the ***java.rmi.server.hostname*** parameter to the real value, and it uses the default of 0.0.0.0. This parameter must be set to the real IP address of the server by manually

editing the **`rhq-server.properties`** file. Restart the server after editing the properties file to load the changes.

Report a bug

# Appendix A. Document History

**Revision 3.3.2-4** **Fri 15 May 2015** **Jared Morgan**
Fixed HTML Anchors that were breaking links in the JBoss ON UI Help > Documentation menu.

**Revision 3.3.2-1** **Tue 28 Apr 2015** **Jared Morgan**
Prepared for JBoss ON 3.3.2 Release

**Revision 3.3.1-6** **Mon Apr 20 2015** **Jared Morgan**
To correct an issue with PDF generation.

**Revision 3.3.1-5** **Fri Feb 27 2015** **Jared Morgan**
Prepared for JBoss ON 3.3.2.GA Release
BZ#1158247 - Corrections to sections containing "run-data-migrator do-it, which is deprecated.
BZ#1164940 - Removed a trailing /bin/java/ from the RHQ_JAVA_HOME path that was causing issues for customers.
BZ#1177772 - Various fixes to rhqadmin/rhqadmin and script names causing issues for customers upgrading.
BZ#1177919 - Corrected instances of the Basic Linux Server setup instructions that suggested that commands needed to be run as root.
BZ#1179421 - Issues with IP addresses declared as "192.68..." instead of "192.168..." fixed globally in the guide.
BZ#1181626 - Corrected Upgrade Instructions that pointed to outdated versions numbers, caused by incorrect XML entities. Additionally cleaned up the upgrade procedure to break up the main concepts into more digestible information chunks.

**Revision 3.3-46** **Thu Dec 11 2014** **Jared Morgan**
Corrected erroneous references to rhqctl.sh in command-line procedures.

**Revision 3.3-45** **Mon Nov 24 2014** **Ella Deon Ballard**
Corrected the upgrade procedures.

**Revision 3.3-42** **Mon Nov 24 2014** **Jared Morgan**
https://bugzilla.redhat.com/show_bug.cgi?id=1162646 - correction to the command to include the shell relevant to rhqctl. Thanks to Ken Hitchcock for this observation.
https://bugzilla.redhat.com/show_bug.cgi?id=1148991 - Added Note to describe the deprecated rhq-encode-password.sh script.